

Vol.6 · No.7 · December 1987

BEEBUG

FOR THE
BBC MICRO &
MASTER SERIES



*Seasons
Greetings*

Snowflakes

-  BEEBUG TRIVIA QUIZ
-  MODULAR ASSEMBLER
-  PERSONAL APPOINTMENTS DIARY
-  Z88 REVIEW

£1.30

FEATURES

Snowflakes	
Pitfall Pete - New Screens	
Procedure and Function Analyser	
Personal Appointments Diary	
Barry Christie Visuals -	
Trench Animation	
Load and Go with Basic Programs	
Exploring Assembler (Part 6)	
The Master Pages -	
ADFS Single Drive Back-up	
Master Hints	
Modular Assembler	
Workshop -	
Thanks for the Memory	
BEEBUG Education	
First Course -	
Colouring your Beeb (Part 2)	
Trivia Quiz	

REVIEWS

6	Cambridge Computer's Z88 -	8
7	a Beeb by any other Name	16
11	A Quest for a Mouse	26
18	Printer Survey (Part 2)	58
	Games Reviews	60
	Word-Ex and Wordwise Plus II	

REGULAR ITEMS

23		4
25		4
29	Editor's Jottings	33-40
	News	45
41	Supplement	68
45	Master Hints	69
46	Hints and Tips	70
	Postbag	71
	Subscriptions & Back Issues	
50	Magazine Disc/Tape	
52		

HINTS & TIPS

GENERAL

54	Easier Game Play
63	Power on Reset
	Getting Input
	Saving Memory when
	using DATA statements
	Which Mode?

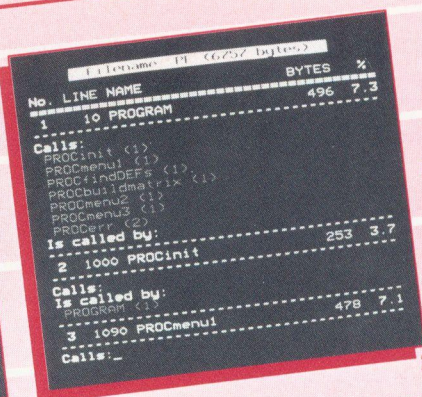
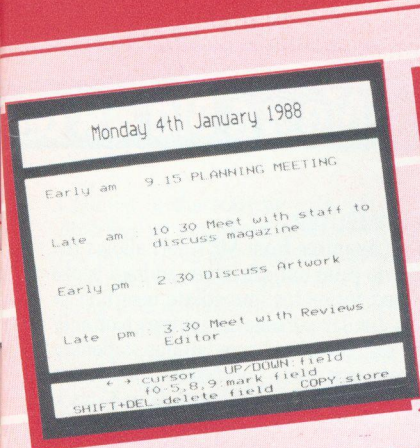
MASTER

	Editorial Rescue
	Direct Screen Access
	Editorial Help
	Master ROM -
	ADT Clash
	Musical Timing
	Appending Basic

PROGRAM INFORMATION

All programs listed in BEEBUG magazine are produced direct from working programs. They are listed in LISTO1 format with a line length of 40. However, you do not need to enter the space after the line number when typing in programs, as this is only included to aid readability. The line length of 40 will help in checking programs listed on a 40 column screen.

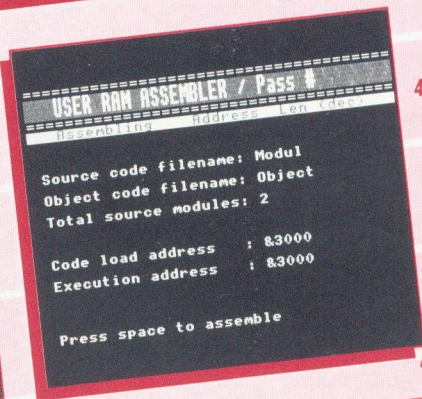
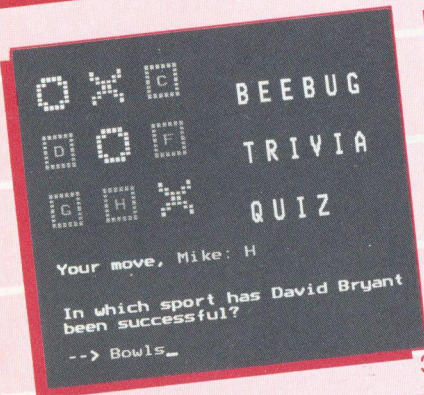
Programs are checked against all standard Acorn systems (model B, B+, Master, Compact and Electron; Basic I and Basic II; ADFS, DFS and Cassette filing systems; and the Tube). We hope that the classification symbols for programs, and also reviews, will clarify matters with regard to compatibility. The complete set of icons is given



1. Personal Appointments Diary

2. Procedure/Function Analyser

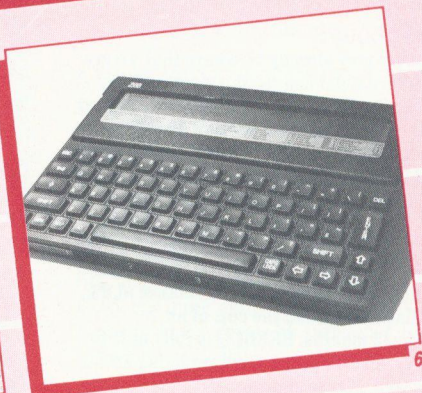
3. Trivia Quiz



4. Modular Assembler

5. Watford Quest Mouse

6. Cambridge Computer's Z88



Computer System

Master (Basic IV)



Compact (Basic VI)



Model B (Basic II)



Model B (Basic I)



Electron



Filing System

ADFS



Dfs



Cassette



Tube Compatibility

Tube



below. These show clearly the valid combinations of machine (version of Basic) and filing system for each item, and Tube compatibility. A single line through a symbol indicates partial working (normally just a few changes will be needed); a cross shows total incompatibility. Reviews do not distinguish between Basic I and II.

Editor's Jottings

CAMBRIDGE COMPUTER'S Z88

This micro, designed by Clive Sinclair, has a surprising amount to offer the BBC micro user, as you will see from our in-depth review in this issue. In fact, so impressed have we been by the quality and power of this highly portable device, that we have arranged a special package deal for BEEBUG Members. Details of this are on page 57.

RISC USER SUBSCRIPTIONS

The response to RISC User, our new magazine for Archimedes users, has been very encouraging, and I would like to remind you of the very favourable rates available to existing BEEBUG members. As such, you can subscribe for the whole of the first year of RISC User for just an additional £6.00 (the introductory rate to non-BEEBUG members is £12.50). The full subscription rate to RISC User will be £14.50 from early 1988, and although we will continue to offer a discounted subscription rate to BEEBUG members, we will not be able to keep to the present level of £6.00

NEW BEEBUG PRODUCT

BEEBUG members may already be aware that we are producing an internal modem for the Master series. This is now awaiting BABT approval which we expect quite soon. Once this is received, we will be able to dispatch to order. This is BEEBUG's first venture into hardware, and one we all feel to be a significant event. This will be followed next year by a modem for the Archimedes.

0% FINANCE FOR AN ARCHIMEDES

Acorn has launched a scheme whereby finance at 0% interest is available to spread the cost of an Archimedes over 12 months. BEEBUG is fully able to offer this facility, and it applies to all trade-ins for an Archimedes as well. Please phone or write for details.

AND FINALLY

This issue of BEEBUG has been produced in record time to ensure dispatch well before Christmas. This will enable us to fulfil all orders received in time for Christmas. We hope you will find plenty to read and to try over the Christmas holiday. We wish you all an enjoyable Christmas and look forward to the New Year.

Mike Williams

News News News Ne

FILOFAX BY COMPUTER

The FAX*FILE Organiser from MEWsoft allows the BBC micro user to print out calendars, address books and user-defined forms for immediate insertion in their Filofax or similar personal organiser. Not only that, MEWsoft can also supply continuous stationery to the correct page size ready punched for insertion in your organiser. FAX*FILE is available on 40 or 80 track disc for £12.95 inc p&p, and a sample of the special paper is included. The paper itself costs £5 for 100 sheets and £20 for 500 sheets, inc. p&p, all from MEWsoft, 11 Cressy Road, London NW3 2NB. Telephone 01-267-2642.

TOPOLOGICAL PRICE TWIST

Topologika, whose adventure games we referred to in our October News spot, has informed us that the prices which they quoted to us were incorrect. The true prices for DFS disc versions of Countdown to Doom, Philosopher's Quest, Kingdom of Hamil and Acheton are £9.95 each (not £17.50). Furthermore, Topologika is offering a 10% discount to BEEBUG members who order direct from them (make sure you quote your membership number). A sequel to Countdown to Doom, to be called Return to Doom, is scheduled for release early next year. Order from Topologika at P.O. Box 39, Stilton, Peterborough PE7 3RL, or telephone (0733) 244682.

CADCAM FOR THE BEEB

Metrotec Ltd has announced improved versions of its EASICAD and EASICAM packages for the BBC micro. EASICAD is a powerful low-cost 2D draughting system, while EASICAM is a user-friendly computer-aided manufacturing system, able to accept geometrical data from EASICAD to produce a part program, with eventual downloading to a CNC machine. Prices range from £138.00 for EASICAD to £224.25 for EASICAM, with various other options. Metrotec Ltd are on Wearside (091) 5109595.

PROLIFIC SUPERIOR

Superior are clearly pulling out all the stops to ensure that Beeb owners have a good choice of games this Christmas. Following releases already covered in our October and November issues, Superior have announced two further games for the Acorn range. Elixir is a bubbling adventure game with Cyril the

Chemist in the starring role, while Spellbinder is described as a captivating 3D arcade adventure set in the mysterious and enchanting Castle of Lorraine. Spellbinder is reviewed in more detail in this issue along with another release from Superior, The Life of Repton. Both of the latest games are available on tape or disc to suite BBC micro, Master, Compact and Electron owners at prices ranging from £9.95 for tape to £14.95 for 3.5" disc. Look in your local computer shop, or contact Superior on Leeds (0532) 459453.

THE MASTER BY ANOTHER NAME

Oak Computers has announced the Mentor range of new low-cost Acorn Master compatible computers for industrial and commercial applications. Systems range from the entry level M2801 at £499 including monitor and disc drive up to the MC2820 with 20MB hard disc and high resolution colour monitor at £1198. For further details phone (0532) 502615.

NOT A PENNY MORE, NOT A PENNY LESS

Domark has made quite a name for itself in the games market since it first released the computer version of Trivial Pursuit last year. Latest releases for the BBC micro from this fast moving company include Countdown, Bullseye, and Treasure Hunt under Domark's new TV Games label - all at £7.95. Star Wars, the computer game, is available on disc only for £12.95, and Not a Penny More, Not a Penny Less, based on the Jeffrey Archer best seller of the same name, is available on cassette for £14.95 and disc for £16.95. All these games should be in the shops now, or contact Domark on 01-947-5622.

ARCHIMEDES

There is plenty of activity on the Archimedes front. The A440 with 4Mbytes of RAM and integral 20Mbyte hard disc system will be available from early December as planned, at prices ranging from £2643.85.

The PC Emulator (software) is out now priced £113.85, but we understand that the hardware version has been postponed indefinitely on Olivetti's orders. The Acorn ROM podule is now at a revised price of £67.85, and Computer Concepts are also producing a ROM podule for £56.35. This will provide 8 ROM/RAM sockets each capable of taking RAM or ROM chips of up to 128K in size, battery-backed as well if required.

Acornsoft has released Logistix, the spreadsheet and time management package at £113.85, and Clares' Image Writer at £29.95 should be out now. Pre-release versions of both products look very promising. In lighter vein, Minerva Systems, better known for its database software, has released a Minotaur game for the Archimedes, and, on a more serious note, its new System DeltaPlus WIMP Card Index at £69.95.

Finally, we have learnt from Acorn that it plans to dispatch the new 1.2 version of the Archimedes operating system to all registered users from 8th December onwards, together with a new Welcome disc, Welcome Guide and a full WIMPS word processor called Arcwriter. If you have an Archimedes, make sure you have completed and returned the registration card to Acorn to receive the free upgrade.

All prices quoted above for Archimedes products include VAT. For more information, the companies mentioned can be contacted as follows:

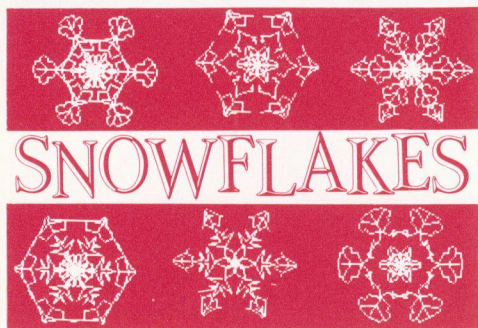
Acorn	(0223) 214411
Clares	(0606) 48511
Computer Concepts	(0442) 63933
Minerva Systems	(0392) 37756

A DABHAND AT SOFTWARE TOO

Dabs Press, best known as publishers of books for the range of BBC micros, had three new software products on show at the Micro User Show in November. Sidewriter is a pop-up menu system for any machine with sideways RAM, MOS Plus provides extensions and improvements to the Master 128 operating system, while the Master Emulation ROM turns the humble BBC micro into a Master look-alike. Prices range from £7.95. For more information telephone 061-773 2413 or write to Dabs Press, 76 Gardener Road, Prestwich, Manchester M5 7HU.

GENIE JUNIOR

PMS, producers of Genie (see review in BEEBUG Vol.6 No.4), has now produced a cut-down software-only version of its professional desk-top pop-up product for the BBC range. Called Genie Junior, this provides the same Genie utilities of address book, desk diary, notepad, calendar and calculator for just £25 (plus £1 p&p). For more information contact PMS on (03552) 32796.



Way back in the winter of '85 we published a picture of a recursively drawn Christmas tree. All that was missing, we said, was the snow. Joan Barnard couldn't resist the challenge, and here is the highly seasonal result.

Snowflakes have a variety of different shapes but all exhibit hexagonal symmetry, since the ice crystals from which they are formed are hexagonal. This program generates a random set of snowflakes, which are similar to real ones, then uses a hardware wrap-around scrolling procedure to simulate snowfall. All you have to do is type in the program, and save it away to disc or tape, before running it and watching the snowflakes fall.

Since the program is capable of generating nearly three hundred different snowflakes, sets of different ones are obtained each time it is run, with duplicates occurring only occasionally.

When typing in the program, the numbering of lines 1600 to 1640, which hold the data, should not be changed as these numbers are referred to directly in PROCflake (lines 1010 to 1030).

Note that after the program has been typed in correctly and seen to be working properly, line 1490 can be changed to:

```
IF ERR=17 THEN GOTO 100
```

This will cause the program to re-run automatically every time Escape is pressed.

PROGRAM NOTES

There are only five different unit shapes used to create the snowflakes, the information needed for drawing these being held in the five DATA lines. These are selected randomly by varying the value of R at lines 1010, 1020, and 1030. Their widths or sizes are also varied by obtaining random values for the variables W and S. The variable D gives the displacement from the centre of the snowflake at which the next layer of units is drawn, its value increasing by fifty for each layer.

If you feel like experimenting with this program more data lines can be added to produce new unit shapes, and lines 1010 to 1030 then altered so that random values of R greater than 1640 are obtained, enabling the new units to be selected on some occasions. The range of values for S and W generated in these lines can also be altered to give some very interesting effects. You can also alter the rate at which the snowflakes scroll on the screen by varying the timing at line 1360.

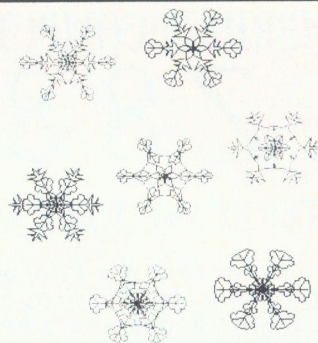
```
10 REM Program Snowflakes
20 REM Version B1.2
30 REM Author Joan Barnard
40 REM Beebug December 1987
50 REM Program subject to copyright
60 :
100 MODE1
110 ON ERROR GOTO 1480
120 VDU23,1,0,0,0,0,0;
130 VDU19,1,6,0,0,0
140 VDU19,2,10,0,0,0
150 VDU19,3,7,0,0,0
160 VDU29,550;860;:PROCflake
170 VDU29,1050;820;:GCOL0,2:PROCflake
180 VDU29,200;600;:PROCflake
190 VDU29,650;512;:GCOL0,1:PROCflake
200 VDU29,1100;420;:GCOL0,3:PROCflake
210 VDU29,250;200;:PROCflake
220 VDU29,750;160;:GCOL0,2:PROCflake
230 PROCscroll
240 END
250 :
1000 DEFPROCflake
1010 X=RND(-TIME):R=RND(3)*10+1590:S=5:
W=RND(5):D=0:PROClayer
1020 R=RND(4)*10+1590:S=RND(2)*10:W=RND
(5)*10:D=50:PROClayer
```



```

1030 R=RND(4)*10+1600:S=RND(3)*10:W=60:
D=100:PROClayer
1040 ENDPROC
1050 :
1060 DEFPROClayer
1070 a=1:b=0:c=0:d=1:PROCs spoke
1080 a=0.5:b=0.866:c=-0.866:d=0.5:PROCs
poke
1090 a=-0.5:b=0.866:c=-0.866:d=-0.5:PRO
Cspoke
1100 a=-1:b=0:c=0:d=-1:PROCs spoke
1110 a=-0.5:b=-0.866:c=0.866:d=-0.5:PRO
Cspoke
1120 a=0.5:b=-0.866:c=0.866:d=0.5:PROCs
poke
1130 ENDPROC
1140 :
1150 DEFPROCspoke
1160 RESTORE R
1170 READ noofpoints
1180 READ x,y:PROCs transform(x,y)
1190 MOVE xt,yt
1200 FOR i=2 TO noofpoints
1210 READ x,y
1220 PROCs transform(x,y)
1230 PLOT 1,xt,yt
1240 NEXT i
1250 ENDPROC
1260 :
1270 DEF PROCs transform(x,y)
1280 xt=a*x+b*y
1290 yt=c*x+d*y
1300 ENDPROC
1310 :
1320 DEFPROCscroll
1330 START=&3000
1340 REPEAT
1350 PROCMOVE(-640)
1360 TIME=0:REPEAT UNTIL TIME>10
1370 UNTIL FALSE
1380 :
1390 DEFPROCMOVE(offset)
1400 START=START+offset
1410 IF START>=&8000 THEN START=START-&
5000

```



```

1420 IF START<&3000 THEN START=START+&5
000
1430 *FX19
1440 VDU23;12,START DIV 2048;0;0;0
1450 VDU23;13,START MOD 2048 DIV 8;0;0;
0
1460 ENDPROC
1470 :
1480 MODE 7:ON ERROR OFF
1490 IF ERR=17:END
1500 REPORT:PRINT" at line ";ERL
1510 END
1520 :
1600 DATA 11,W,D,5,5,-5,5,10,5,-W,15,-1
0,20,-10,-20,-W,-15,10,-5,-5,-5,5,-5
1610 DATA 23,S,D+5,0,15,10,0,5,10,-5,10
,-10,0,-S,-20,S,20,0,10,-S,5,-S,-5,0,-10
,S,-20,-S,20,-10,0,-5,-10,5,-10,10,0,0,-
15,S,-10,0,50,0,-50,S,10
1620 DATA 19,0,10+D,S,-10,10,20,S,-S/2,-
S,S/2,-S,20,-5,10,-10,0,-5,-10,-S,-20,-S
,-S/2,S,S/2,10,-20,S,10,0,40,0,-15,-S,-1
0,S,10,S,-10
1630 DATA 13,5,D,30,20,-30,-5,0,10,20,1
0,-20,-5,-5,20,-5,-20,-20,5,20,-10,0,-10
,-30,5,30,-20
1640 DATA 13,W,D,-W/2,30,-15,-30,15,30,
-W/2,20,-W/2,-20,15,-30,-15,30,-W/2,-30,
10,0,W-10,40,W-10,-40,10,0

```



PITFALL PETE

NEW GAMES SCREENS FOR CHRISTMAS



Last Christmas we published a game called Pitfall Pete which proved highly popular (Vol.5 No.7). We have persuaded the author, Jonathan Temple, to provide a further 10 screens for this enjoyable game, and these are included as an additional item on this month's disc/tape. Unfortunately, it is not feasible to print the data for these extra screens in the magazine. Full instructions for adding these screens to your original copy of Pitfall Pete are also provided on the disc/tape.

CAMBRIDGE COMPUTER'S Z88

- A BEEB BY ANY OTHER NAME ?

Disappointed that his Filofax is not compatible with his BBC micro, Sheridan Williams tries out the Cambridge Computer Z88 in the same role.

Product	Z88 Portable Computer
Manufacturer	Cambridge Computer Ltd Freepost, Cambridge CB4 1BR. Tel. (0223) 312216
Price	£287.50 inc VAT

In BEEBUG Vol.5 No.10, we reviewed the Psion Organiser because of its usefulness when allied to the BBC (& Master) computer. However, being designed to fit the pocket, the Psion Organiser (even in Organiser II guise) suffers from the dual limitations of screen and keyboard size. The Z88 overcomes both of these limitations as far as is practicable in a portable computer designed to fit easily in a briefcase, though too big for the pocket. The Z88 is amazingly compatible with the BBC micro because it contains BBC Basic together with a word processor / spreadsheet / database package called Pipedream which is identical to View Professional on the BBC.

The Z88 is the size of an A4 sheet of paper but 25mm thick, and weighs just 950g. The screen shows 8 lines at full A4 width, with a maximum of 93 visible characters per line. The QWERTY keyboard has a pretty good feel to it considering the cost of the system. An audible keyclick can be turned on or off at will, and is useful in counteracting the otherwise dead feel of the keys. The whole unit can be tilted with an extendable flap under the machine, and the contrast of the screen is adjustable. Files are stored in RAM or EPROM, although there are rumours that a disc drive will be available shortly. Power is provided by four AA type batteries, although a mains power supply is also available. Depending on the number of

RAM and EPROM packs installed, you have between 1 and 6 minutes to replace the batteries before losing all your files in RAM (files in EPROM are not lost).

The Z88 is switched on by pressing both the shift keys simultaneously, and the screen will show whatever you were working on last. Three keys found at the bottom left of the keyboard, marked INDEX, MENU and HELP will select all the in-built software packages, options within packages, and any help that you may require. Alternatively, all packages and options have their own individual key sequences to access them, provided you can remember what these are. Cambridge quite rightly assume that you will be using Pipedream (of which more later) most frequently, and endow the fascia with a strip between the screen and the keyboard showing the most frequently used commands.



Working round the machine starting from the left of the keyboard there is a wheel to adjust the screen contrast, and a socket for the mains adaptor. On the right of the keyboard is a 9-pin D-type socket: this is a serial port, and is used to communicate with the outside world (including a BBC micro). The modem, printer and connections to other computers use this port. I was confused that Cambridge Computer have a parallel printer lead on their list of peripherals when there is no parallel port to connect it to! However, having obtained one and examined it, I found that it contains a black box which converts from serial to parallel, which also helps to explain why it costs £29.95.

Continuing our tour round the machine we next encounter a removable plastic cover concealing a 20-way edge connector, presumably for connecting a disc drive when developed. Finally, on the front of the Z88 under a transparent cover, there are three empty sockets for ROM/RAM packs. Cambridge Computer lists 32k RAM, 32k ROM, 128k RAM and 128k ROM packs amongst their accessories, and as far as I can see, any combination of packs may be inserted into any of the sockets. As all storage takes place on RAM and EPROM, I took advice from an existing user, and ordered a 128k RAM pack with my Z88.

The Z88 comes with a 217 page manual. This is very well written, with a comprehensive index, and contains several good examples. It does however, leave a few questions unanswered, for example, I still have not found out how to divert output from a Basic program to the printer. It states how to LIST the program to the printer, but that's not what I want.

THE PANEL

The Z88 is very flexible when it comes to setting up various options. Using the "Panel" you may change the following:

Keyclick On/Off	Insert/Overtyp
Keyboard Auto Repeat	Sound Yes/No
Default Device	Default Directory
Machine Timeout	Map on, Map size
Transmit baud rate	Receive Baud Rate
Parity	Xon/Xoff
Date format - European/USA	

One of the Panel options is Map, an ingenious, and very useful feature displayed on the right hand side of the screen. This shows, in dot form, the actual layout of the document on which you are working (remember that you are working on a maximum of 7 visible lines of text only, the other line contains the cursor's current row and column coordinates).

The Z88 could be, for many BBC and Master owners, the passport to a previously unknown freedom. They can use many of the features of their computer, but in any environment - train, lounge, car or wherever. The major features that ally the Z88 to the BBC computer are its

BBC Basic and Pipedream - which is, in reality, "View Professional".

Also on-board, and instantly available from within any other application, are a diary with alarm, calendar, clock, calculator, printer-driver editor, comprehensive ADFS-style filing system, terminal emulator, and an import/export facility allowing easy transfer between computer systems, especially the BBC & Master.



USING THE Z88

I found the Z88 enormously useful. For those making long journeys with time to spare, the Z88 offers superb flexibility, and in many instances would be worth purchasing for that application alone. The more I use the Z88, the more useful I find it to be; and as with any sophisticated system, I am constantly finding new features.

PIPEDREAM

Pipedream is virtually identical to View Professional, and I therefore suggest that you read our review of View Professional in BEEBUG Vol.6 No.4 for a full description, suffice to say we were impressed. In a nutshell, Pipedream combines the features of a wordprocessor, spreadsheet, and database all in one, and once mastered, forms a superbly co-ordinated system.

Because of the compatibility between Pipedream on the Z88, and View Professional on the BBC, the transfer of documents could not be simpler. It is a shame (and a distinct

disadvantage) therefore, that the commands in Pipedream and View Professional are so different. For example, to re-format a paragraph in Pipedream you give the command R, whereas in View Professional you use FP. Virtually no commands are the same, so you need to learn two sets. Remember though, that all commands are available in Pipedream from a menu should you forget them, but this is not as fast and convenient as giving the commands directly. Considering that both packages were written by Colton Software I find this incompatibility difficult to comprehend. Despite this, Pipedream makes the Z88 worth buying for this feature alone, whether you are a BBC micro user or not.

The Z88 should not be dismissed as being a machine only capable of handling a small amount of text. The basic Z88, together with a single 128k RAM pack, (which I suggest to be a working minimum configuration) will hold over 21,000 words, and Pipedream has facilities for linking many files to form a larger document. Assuming that you have equipped the machine with three 128k RAMs, then together with its internal 32k memory, the Z88 will allow the storage of around 70,000 words!

To test Pipedream's performance when sorting, I created a 7 field database with 128 name and address records. The time to sort the file was 16 minutes, which is not very good. However, when adding records into a name and address file you would be better advised to insert them in the correct position, avoiding the need for a re-sort on every insertion.

This review was written on the Z88, and contains around 2,300 words. Towards completion of the article the speed of key entry began to be impaired during insertion of text. However, simple addition of text at the end of the article was not affected, even though I had the map switched on. The manual says that switching off the map results in an increase in the speed of response.

PRINTER EDITOR

Any style of which your printer is capable, can be used during the printing of Pipedream documents. For example underline, bold, italics, superscripts, subscripts, proportional

spacing, condensed etc. In the first three instances Pipedream will actually show the effect on the screen.

The Z88 comes with a built-in printer driver which interfaces with the Epson (and Epson compatible) series of printers. The Printer Editor allows you to create your own printer driver definition, and store any number of such definitions. The Printer Editor is superbly flexible and easy to use, and I cannot think of a printer for which a driver could not be created. Even backspace to underline is easy to implement.

BBC BASIC

Yes, it is truly BBC Basic, and as far as I can tell it appears to be BASIC IV, because the LIST IF command is implemented (although LIST IF is not mentioned in the manual). It was written by Richard Russell who wrote Z80 Basic for the BBC micro. All the BBC Basic commands and features that are appropriate to a small screen system are supported. Those that are not are the graphics and sound command sets. Also, although star commands are not supported OSCLI is, although I could not get it to operate. I had no problem in transferring any of my non-graphics programs across from the BBC, and all worked first time. Transfer is not a direct process, but requires a spool file to be created and sent to the Z88. Once in the Z88 you can use Pipedream to edit in some leading commands and then execute it into Basic (the filing system has an execute option that operates like *EXEC on the Beeb). Quite straightforward once you are used to the process.

BBC Basic on the Z88 is not as fast as on the BBC micro, but it compares reasonably well with other micros, and I never found it to be too slow in actual practice. For applications requiring the manipulation of files however, the Z88 is a good deal faster than the BBC micro. This is because the Z88's file storage is solid state, whereas the Beeb's file storage is on floppy disc. One disappointing omission is that of a line editor, even to the extent of their being no cursor copying either. If you mis-type a Basic line you will have to re-type the entire line to correct it (GROAN), thus you soon learn to steer clear of multi-statement lines.

continued on page 57

PROCEDURE AND FUNCTION ANALYSER

This helpful utility by Graham Crow will analyse any Basic programs and provide a comprehensive cross-reference listing of all the functions and procedures used. Altogether, a most useful utility for all Basic programmers.

BBC Basic encourages 'structured' programming, by providing procedures and functions. These allow self-contained program modules to be written and called by name, either from the main part of the program, or from within another procedure or function, or even (recursively) from within themselves. Well-structured programs tend to comprise a fairly short control section, followed by a number of procedures and functions.

This utility helps to clarify the structure of a program (previously saved to disc or tape) by identifying each procedure or function, and listing all the procedures or functions that it calls, together with all those by which it is called. The line number of each DEF statement is given, together with the length of the procedure or function in bytes, and its length as a percentage of the whole program. The listing may be to screen or printer (Epson-compatible), and may either be limited to the DEF lines, or include all calls. It is invaluable for examining the structure of your own programs, and (perhaps more importantly) for analysing other people's.

The program is suitable for all models of the BBC micro, including the Master series. Because it is written in mode 7, however, it will not run on the Electron.

HOW TO USE THE PROGRAM

To use the utility (called PFREF), you will need to type in the listing and save to disc or tape.

When you run the program, choose either 'DEF lines only' or 'DEF lines and all calls', from the main menu. The latter gives more information, but is consequently slower.

Make sure that the source disc, containing the target program to be analysed, is in the appropriate drive, and enter the filename of this program when prompted. As the program runs, the line numbers of the target program are displayed to confirm progress. If you have chosen option 2, the program makes two passes through the target program. The target program can also be analysed from tape, but you will have to rewind the tape before the second pass can proceed (or record the target program twice).

Filename: PFREF (6760 bytes)

No.	LINE	NAME	BYTES	%
1	10	PROGRAM	499	7.4
2	1000	PROCinit	253	3.7
3	1090	PROCmenu1	478	7.1
4	1290	PROCtitle	108	1.6
5	1410	PROCindDEFs	452	6.7
6	1510	PROCkeybytes	317	4.7
7	1510	PROCinTable	207	3.1
8	1570	FNitemname	186	2.8
9	1620	PROCbuildmatrix	311	4.6
10	1730	PROCinMatrix	101	1.5
11	1760	FNnum	123	1.8
12	1810	PROCmenu2	240	3.6
13	1880	PROCmenu3	386	5.7
14	1990	PROCprinter	286	4.2
15	2060	PROCscreenDEFs	127	1.9
16	2110	PROCscreenhead	226	3.4
17	2170	PROCscreenline	162	2.4
18	2220	PROCsetinfo	128	1.9
19	2260	PROCscreencalls	366	5.4
20	2360	FNname	82	1.2

Simple procedure/function listing to screen

When the analysis of the program is complete, you are presented with a second menu, which allows you to select either 'DEF lines only' or 'DEF lines and all calls' to be displayed (depending on your initial choice). You may then choose whether the output should be to the screen or the printer. Screen output is in paged mode, so press Shift at the end of each screenful. Printed output is for Epson-compatible printers. To return to the menu, press Escape at any time during the display, or the space bar once it is complete.

PROGRAM NOTES

The program is written entirely in Basic and occupies about 7940 bytes. There are four sections to the program.

THE CONTROL SECTION

The control program comprises lines 100-120. Line 110 makes a call to PROCmenu1 (line 1090) which displays the main menu, and opens the file containing the target program. Depending on the response, the flag 'both' is set to TRUE or FALSE. PROCfindDEFs scans the target program and builds up a table starting at address &4000 of procedure and function names and line numbers. IF 'both' is FALSE, control is passed to line 170 and at line 180 a continuous REPEAT-UNTIL loop displays the final menu. IF 'both' is TRUE, two addresses (M% and E%) are calculated in lines 140 and 150, defining the limits of a matrix. PROCbuildmatrix makes a second pass through the target program and places in the matrix details of all the procedure and function calls. In this case, line 190 sets up an alternative final menu. Both versions of the final menu provide an exit from the program.

THE FIRST PASS

During the first pass, the program is read in a byte at a time, and examined for the DEF token. When found, the procedure or function name is extracted (along with the line on which it occurs) and stored in an array.

THE SECOND PASS

Again the file is examined a byte at a time, this time looking for any calls to procedures or functions. When found, the name of the procedure or function is extracted, and cross-referenced with a list of those compiled during the first pass. Details of these are then stored in another array for later use.

THE PRINT OPTIONS

The last section of the program deals with the closing menus and the print options which they offer. Line 170

alters the ON ERROR so that if Escape is pressed, control is passed to the appropriate menu loop. If the user has requested 'DEF lines and all calls' at the opening menu, then this option is also available in the closing menu (PROCmenu3 - line 1880); otherwise, only the DEF lines may be printed (PROC- menu 2 - line 1810).

PROCprinter (line 1990) determines whether output is to be to the printer or the screen. If the printer is to be used, then the flag 'printer' is set, and condensed mode selected at 8 lines per inch, with the screen disabled using:

VDU2,1,15,1,27,1,48,21

```
PROC/FN ANALYSER
  by Graham Crow

1...DEF lines only
2...DEF lines and all calls
3...Exit

Which? 2

Insert the disc containing
the target program.

Enter filename..... PFREF
First pass Line... 1090
```

Initial dialogue screen

NO.	LINE	NAME	BYTES	%	:	CALLS	:	IS CALLED BY
1	10	PROGRAM	456	6.5	:	2. PROCinit (1)	:	
					:	3. PROCmenu1 (1)	:	
					:	5. PROCfindDEFs (1)	:	
					:	9. PROCbuildmatrix (1)	:	
					:	12. PROCmenu2 (1)	:	
					:	13. PROCmenu3 (1)	:	
					:	27. PROCerr (2)	:	
2	150	PROCinit	261	3.7	:		:	1. PROGRAM (1)
3	240	PROCmenu1	584	8.4	:	4. PROCtitle (1)	:	1. PROGRAM (1)
					:	26. PROCcscli (1)	:	
					:	28. PROCcodb (1)	:	
					:	29. PROCexit (1)	:	

Example of full listing to printer


```

10 REM Program PFREF
20 REM Version B1.47
30 REM Author Graham Crow
40 REM Beebug December 1987
50 REM Program subject to copyright
60 :
100 MODE 7:ON ERROR PROCerr
110 PROCinit:PROCmenu1
120 VDU31,3,18,136,c:IF both PRINT"Fir
st pass" ELSE PRINT"Processing"
130 VDU31,17,18,137:PRINT"Line..."w$:P
ROCfindDEFs:IF NOT both THEN 170
140 M%=T%+(p%-T%):REM start of Matrix
150 E%=M%+C%^2:REM end of Matrix
160 VDU31,3,18,136,c:PRINT"Second pass
":PRINTTAB(26,18)SPC(6):PROCbuildmatrix
170 ON ERROR IF ERR=17 THEN 180 ELSE I
F ERR<>17 PROCerr
180 VDU6,3,23,1,1;0;0;0;:IF NOT both R
EPEAT PROCmenu2:UNTIL FALSE
190 REPEAT PROCmenu3:UNTIL FALSE
200 :
1000 DEFPROCinit:CLOSE#0:VDU2,1,27,1,64
,3:*FX4,1
1010 T%=&4000:REM Table of P/F names &
DEF lines
1020 m%=20:REM max length of P/F name
1030 C%=1:REM count of P/Fs
1040 g=130:c=134:y=131:w=135:d=141:REM
Control codes
1050 g$=CHR$g:c$=CHR$c:y$=CHR$y:w$=CHR$
w
1060 t1=10:t2=60:t3=95:REM tabs
1070 ENDPROC
1080 :
1090 DEFPROCmenu1:PROCTitle
1100 VDU31,4,6,c:PRINT"1...DEF lines on
ly"
1110 VDU31,4,7,c:PRINT"2...DEF lines an
d all calls"
1120 VDU31,4,8,c:PRINT"3...Exit":VDU31,
4,10,y:PRINT"Which?"w$;
1130 REPEAT G=GET-48:UNTIL G>0 AND G<4:
VDU G+48
1140 IF G=3 PROCexit:ELSE IF G=2 both=
TRUE ELSE both=FALSE
1150 VDU31,4,12,c:PRINT"Insert the disc
containing"
1160 VDU31,4,13,c:PRINT"the target prog
ram."
1170 VDU31,4,16,y:PRINT"Enter filename.
...."w$;:INPUTTAB(26,16)""file$
1180 VDU23,1,0;0;0;0;:IF file$=""VDU7:G
OTO 1170
1190 IF file$=""VDU7:GOTO 1170
1200 F%=OPENUP(file$):IF F%=0 PROCboob(
"No such file!"):GOTO 1170
1210 ENDPROC

```

Filename: PFREF (6760 bytes)			
No.	LINE	NAME	BYTES %
=====			
Calls:			
Is called by:			
PROCmenu1 (1)			
PROCmenu2 (1)			
PROCmenu3 (1)			

5	1270	PROCfindDEFs	452 6.7
Calls:			
PROCkeybytes (2)			
PROCinTable (1)			
FNitemname (1)			
Is called by:			
PROGRAM (1)			

6	1410	PROCkeybytes	317 4.7
Calls:			
PROCboob (1)			

Example of full listing to screen

```

1220 :
1230 DEFPROCTitle:VDU3,26,12
1240 FOR J%=1 TO 2:VDU31,9,J%,d,y:PRINT
"PROC/FN ANALYSER":NEXT
1250 VDU31,11,3,g:PRINT"by Graham Crow"
:ENDPROC
1260 :
1270 DEFPROCfindDEFs
1280 REM first line
1290 p%=T%:REM pointer
1300 Q%=2:REM length of P/F (starts wit
h 2 end bytes)
1310 H%=0:REM program length
1320 PROCkeybytes:name$="PROGRAM":$p%=n
ame$:p%=p%+LEN(name$)+1
1330 ?p%=L% DIV 256:p%=p%+1:?p%=L% MOD
256:p%=p%+1:REM line hi/lo
1340 C%=C%+1:end=FALSE:REPEAT:PTR #F%=P
%+len%:REM goto next line
1350 PROCkeybytes
1360 IF B%=&DD REPEAT B%=BGET#F%:UNTIL
B%<>32:PROCinTable(FNitemname)
1370 UNTIL end
1380 ?p%=Q% DIV 256:p%=p%+1:?p%=Q% MOD
256:p%=p%+1:REM P/F length hi/lo
1390 H%=H%+Q%:C%=C%-1:ENDPROC
1400 :
1410 DEFPROCkeybytes:P%=PTR#F%:REM poin
ter=start of line
1420 REM read first 4 bytes
1430 B%=BGET #F%:IF B%<>13 PROCboob("No
t a BASIC program!"):RUN
1440 B%=BGET #F%:IF B%>=&7F end=TRUE:EN
DPROC
1450 L%=B%*256:B%=BGET#F%:L%=L%+B%:REM
line no
1460 PRINT:TAB(26,18)L%:B%=BGET #F%:le
n%=B%:REM length
1470 REPEAT B%=BGET #F%:UNTIL B%<>32

```



```

1480 IF B%<>DD THEN Q%=Q%+len%:REM fir
st non-space byte
1490 ENDPROC
1500 :
1510 DEFPROCinTable(name$)
1520 ?p%=Q% DIV 256:p%=p%+1:?p%=Q% MOD
256:p%=p%+1:REM P/F length hi/lo
1530 $p%=name$:p%=p%+LEN(name$)+1
1540 ?p%=L% DIV 256:p%=p%+1:?p%=L% MOD
256:p%=p%+1:REM line hi/lo
1550 C%=C%+1:H%=H%+Q%:Q%=len%:ENDPROC
1560 :
1570 DEFFNitemname:IF B%=&F2 N$="PROC"
ELSE N$="FN"
1580 REPEAT:B%=BGET #F%:N$=N$+CHR$(B%)
1590 UNTIL B%<48 OR (B%>57 AND B%<65) O
R (B%>90 AND B%<95) OR B%>122
1600 N$=LEFT$(N$,LEN(N$)-1):=LEFT$(N$,m
%):REM truncate name if necessary
1610 :
1620 DEFPROCbuildmatrix:FOR J%=M% TO E%
STEP 4:J%=0:NEXT
1630 end=FALSE:PTR #F%=0:I%=1:REM curre
nt P/F
1640 B%=BGET #F%:REPEAT:B%=BGET #F%:IF
B%=&7F end=TRUE:GOTO 1710
1650 L%=B%*256:B%=BGET #F%:L%=L%+B%:PRIN
T;TAB(26,18)L%;
1660 B%=BGET #F%:REM length
1670 REPEAT B%=BGET #F%
1680 IF B%&DD REPEAT B%=BGET #F%:UNTIL
B%<>32:N$=FNitemname:I%=I%+1
1690 IF B%=&F2 OR B%=&A4 PROCinMatrix
1700 UNTIL B%=&D
1710 UNTIL end:ENDPROC
1720 :
1730 DEFPROCinMatrix:N$=FNitemname:N%=F
Nnum(N$):IF N%>C% ENDPROC
1740 cell%=(I%-1)*C%+N%:A%=M%+cell%-1:?
A%=?A%+1:ENDPROC
1750 :
1760 DEF FNnum(name$):p%=T%:c%=1:found=F
ALSE
1770 REPEAT:IF $p%=name$ THEN found=TRU
E:GOTO 1790
1780 p%=p%+LEN($p%)+5:c%=c%+1
1790 UNTIL found OR c%>C%:=c%
1800 :
1810 DEFPROCmenu2:PROctitle:VDU31,10,5,
g:PRINT"Filename:"w$;file$
1820 VDU31,4,9,c:PRINT"1....List DEF li
nes":VDU31,4,10,c:PRINT"2....Exit"
1830 VDU31,4,12,y:PRINT"Which?"w$;
1840 REPEAT:G=GET-48:UNTIL G>0 AND G<3:
VDU G+48:IF G=2 PROCexit
1850 PROCprinter:IF printer PROCprintDE
Fs ELSE PROCscreenDEFs
1860 VDU6:ENDPROC

```

```

1870 :
1880 DEFPROCmenu3:PROctitle:VDU31,10,5,
g:PRINT"Filename:"w$;file$
1890 VDU31,4,9,c:PRINT"1....List DEF li
nes only"
1900 VDU31,4,10,c:PRINT"2....List DEF l
ines and all calls"
1910 VDU31,4,11,c:PRINT"3....Exit":VDU3
1,4,13,y:PRINT"Which?"w$;
1920 REPEAT:G=GET-48:UNTIL G>0 AND G<4:
VDU G+48:IF G=3 PROCexit
1930 PROCprinter:IF G=1 AND printer PRO
CprinterDEFs
1940 IF G=1 AND NOT printer PROCscreenD
EFs
1950 IF G=2 AND printer PROCprintcalls
1960 IF G=2 AND NOT printer PROCscreenC
alls
1970 VDU6:ENDPROC
1980 :
1990 DEFPROCprinter:VDU31,4,16,y:PRINT"
PRINTER? (Y/N):"w$;G$=GET$:PRINT G$
2000 IF G$<>"Y" printer=FALSE:ENDPROC:E
LSE printer=TRUE
2010 VDU31,4,18,y:PRINT"PREPARE PRINTER
- PRESS ANY KEY ";:get=GET
2020 VDU31,4,18:PRINT SPC(40)
2030 IF get=42 VDU31,4,18,y,42:INPUT""F
$:PROCoscli(F$):get=0:GOTO 2010
2040 VDU31,3,18,136,c:PRINT"Printing ";
:VDU2,1,15,1,27,1,48,21:ENDPROC
2050 :
2060 DEFPROCscreenDEFs:PROCscreenhead:V
DU14:ptr%=T%
2070 FOR J%=1 TO C%:PROCscreenline:NEXT
2080 PRINT STRING$(39,"=")'"Press SPAC
E BAR ";
2090 REPEAT UNTIL GET=32:VDU15:ENDPROC
2100 :
2110 DEFPROCscreenhead:lg%=LEN(file$)+L
EN(STR$(H%))+22:lg%=lg%DIV2
2120 tab%=18-lg%:VDU12,31,tab%,0,y,157,
132
2130 PRINT"Filename: "file$;" ("H%;" b
ytes) "CHR$(156)
2140 VDU31,0,2:PRINT"No."c$"LINE"y$"NAM
E"SPC(15)c$"BYTES"y$" %"
2150 PRINT STRING$(39,"="):VDU28,0,24,3
9,4:ENDPROC
2160 :
2170 DEFPROCscreenline:PROCgetinfo:PROC
numformat(J%,0,2)
2180 VDUc;:PROCnumformat(L%,0,5):VDUy;:
PROCstrformat(N$,0,m%)
2190 VDUc;:PROCnumformat(Q%,0,4):VDUy;:
PROCnumformat(Q%*100/H%,1,4)
2200 PRINT:ENDPROC
2210 :

```



```

2220 DEFPROCgetinfo:N$=$ptr%:ptr%=ptr%+
LEN($ptr%)+1
2230 L%=(?ptr%*256)+(ptr%?1):ptr%=ptr%+
2:Q%=(?ptr%*256)+(ptr%?1):ptr%=ptr%+2
2240 ENDPROC
2250 :
2260 DEFPROCscreencalls:PROCscreenhead:
VDU14:ptr%=T%
2270 FOR J%=1 TO C%:PROCscreenline:PRIN
T STRING$(39,"-"):PRINT"Calls:"
2280 FOR N%=1 TO C%:cell=(J%-1)*C%+N%:A
%=M%+cell-1:IF ?A%=0 THEN 2300
2290 PRINT CHR$(130)+FNname(N%) " (" ;?A%
;" )"
2300 NEXT:PRINT "Is called by:"
2310 FOR N%=1 TO C%:cell=(N%-1)*C%+J%:A
%=M%+cell-1:IF ?A%=0 THEN 2330
2320 PRINT CHR$(130)+FNname(N%) " (" ;?A%
;" )"
2330 NEXT:PRINT STRING$(39,"-")
2340 NEXT:PRINT""Press SPACE BAR " ;:RE
PEAT UNTIL GET=32:VDU15:ENDPROC
2350 :
2360 DEFFNname(num%):p%=T%:IF num%=1 TH
EN =$p%
2370 FOR K%=2 TO num%:p%=p%+LEN($p%)+5:
NEXT:=$p%
2380 :
2390 DEFPROCprintDEFS:PRINT'TAB(t1)"Fi
lename: "file$ " (" ;H% ;" bytes)"
2400 PRINT'TAB(t1)STRING$(42,"=")
2410 PRINTTAB(t1)"No. LINE NAME"SPC(18
)"BYTES %"
2420 PRINTTAB(t1)STRING$(42,"=")
2430 ptr%=T%:FOR J%=1 TO C%:PROCprintli
ne:PRINT:NEXT
2440 PRINTTAB(t1)STRING$(42,"="):VDU6,3
:ENDPROC
2450 :
2460 DEFPROCprintline:PROCgetinfo:PROCn
umformat(J%,0,t1+2)
2470 PROCnumformat(L%,0,6):PROCstrforma
t(N$,2,m%):PROCnumformat(Q%,0,6)
2480 PROCnumformat(Q%*100/H%,1,6):ENDPR
OC
2490 :
2500 DEFPROCprintcalls:PRINT'TAB(t1)"Fi
lename: "file$ " (" ;H% ;" bytes)"
2510 PRINT'TAB(t1)STRING$(120,"=")
2520 PRINTTAB(t1)"No. LINE NAME"SPC(18
)"BYTES %";
2530 PRINTTAB(t2-3)": "TAB(t2+5)"CALLS"TA
B(t3-3)": "TAB(t3+5)"IS CALLED BY"
2540 PRINTTAB(t1)STRING$(120,"="):ptr%=
T%:FOR J%=1 TO C%:PROCprintline
2550 colend=FALSE:rowend=FALSE:colempty
=TRUE:rowempty=TRUE:c%=0:r%=0:@%=2
2560 REPEAT:PROCrow:PROCcol

```

```

2570 IF NOT(colend AND rowend) PRINTTAB
(t2-3)": ";
2580 IF NOT rowend PRINTTAB(t2)c% ;". "
FNname(c%) " (" ;?rad% ;" )";
2590 IF NOT(colend AND rowend) PRINTTAB
(t3-3)": ";
2600 IF NOT colend PRINTTAB(t3)r% ;". "
FNname(r%) " (" ;?cad% ;" )";
2610 UNTIL rowend AND colend
2620 IF colempty AND rowempty PRINTTAB(
t2-3)": "TAB(t3-3)": ";
2630 @%=10:PRINT'TAB(t1)STRING$(120,"-
");NEXT:VDU6,3:ENDPROC
2640 :
2650 DEFPROCrow
2660 REPEAT:c%=c%+1:cell=(J%-1)*C%+c%:r
ad%=M%+cell-1:UNTIL ?rad%>0 OR c%>C%
2670 IF c%>C% rowend=TRUE
2680 IF ?rad%>0 rowempty=FALSE
2690 ENDPROC
2700 :
2710 DEFPROCcol
2720 REPEAT:r%=r%+1:cell=(r%-1)*C%+J%:c
ad%=M%+cell-1:UNTIL ?cad%>0 OR r%>C%
2730 IF r%>C% colend=TRUE
2740 IF ?cad%>0 colempty=FALSE
2750 ENDPROC
2760 :
2770 DEFPROCoscli(F%):$&700=F$:X%=0:Y%=
7:CALL &FFF7:ENDPROC
2780 :
2790 DEFPROCerr:CLOSE #0:@%=10:VDU7,15,
6,3,26,12,23,1,1;0;0;0;:*FX4
2800 REPORT:PRINT" at line " ;:ERL:END
2810 :
2820 DEFPROCboob(boob$):VDU23,1,1;0;0;0
;
2830 VDU7,15,31,3,18,136,c:PRINT boob$+
STRING$(40," ")
2840 VDU31,4,20,c:PRINT "Press Q to Qui
t or R to RUN ";
2850 REPEAT G$=GET$:UNTIL G$="Q" OR G$=
"R":IF G$="R" THEN RUN:ELSE PROCexit
2860 :
2870 DEFPROCexit:CLOSE#0:@%=10:VDU6,3,1
5,26,12:*FX4
2880 END
2890 :
2900 DEFPROCnumformat(num,places%,field
%)
2910 IF places%=0 @%=field%:GOTO2930
2920 @%=$20000+places%*256+field%
2930 PRINT num;:@%=10:ENDPROC
2940 :
2950 DEFPROCstrformat(str$,gap%,field%)
2960 PRINT SPC(gap%);str$;SPC(field%-LE
N(str$));:ENDPROC

```




Roger Burg, a professional computer graphics artist of some repute, takes a look at Watford Electronics' most recent "Art" package.

Product Supplier	Quest Mouse and Quest Paint Watford Electronics 250 High Street, Watford WD1 2AN. Tel. (0923) 37774
Price	£70.15 inc VAT and p&p Quest Font Disc £13.80 extra

'Another mouse package' I thought - but this one held several pleasant surprises, and contained interesting improvements on earlier artistic rodents. Installing the ROM unit and the mouse was straightforward, and once the sick mouse had been diagnosed, (Watford Electronics replaced it at once) the package was instantly available with a couple of star commands.

The drawing screen uses text menus crushed into boxes, and the redefined mode 1 text cleverly simulates 80 columns by using special narrow letters. I would have preferred the clarity and compactness of icons, which would also have helped sell the package to non-English speaking countries.

The program partly adapts itself to the available memory of the machine, and uses back-up screens in most instances to allow 'undo' commands. It works with both ADFS and DFS, and is not dependent on the Acornsoft graphics ROM. At last mode 1 seems to have been adopted as the mode for most serious graphics, though I was disappointed to find that no other mode was available, not even to reclaim extra memory.

The program is crammed full of functions; too many to list intelligibly. All the elementary colour, drawing, line, arc, solid and outline ellipses, and dotted line functions (which one would expect) are present and are well implemented at an excellent speed for interactive art.

There are various brushes, a colour cycle including user-defined options, colour sensitive cut-and-paste between pictures, with rotations, reflections and scaling, grid-lockable patterns, dither colours, flood and other fills. The appropriate features have user-definable types or parameters, and nearly all features interact most consistently and intelligently.

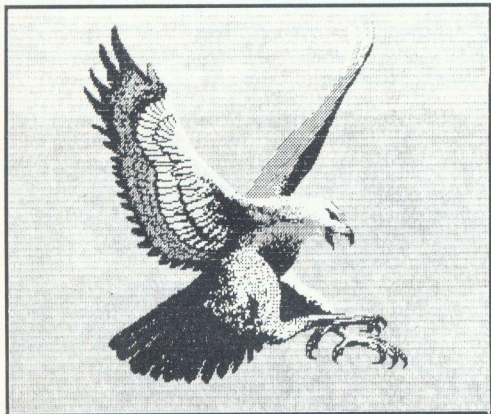
Any reservations about the package must be read in the light of the quality and range of the majority of the program. Especially welcome is the use of lines extending a proportion of the way towards a vanishing point, though it is still not possible to place these vanishing points off the screen.

Ellipses can be 'stretched' as you draw, either vertically or horizontally. It takes only one two-byte multiplication to make these rotatable. But any attention paid to the artist is welcome!

The user can select colour-plotting mode, and this has been extended to plot conditionally, depending on the colour of the pixel to be changed. This gives the effect of being able to draw 'over' or 'under' other colours, and the range of logical colours considered is entirely under the user's control.

The air brush was implemented particularly well, giving a good random coverage, with a suitable increase in density toward the centre. A limited range of speed avoids the tendency to obliterate hours of detail with a second of solid colour, though, as is so often the case when good routines are possible in Basic, it is irritating that 32K of machine code cannot do anything other than spray a perfectly square area. Still, enough carping! The zoom is particularly easy to use, and follows the trend of using an increasingly small, on-screen zoom window, with excellent scrolling.

Quest Mouse does not support output to a spooled file, which would be a useful facility allowing transfer to other screen modes. However, it does allow pictures larger than screen size by automatic disc shuffling. The program has marked similarities to Cheese, from the same programmer Ian Bradbury, who I understand is specialising in excellent quality graphics packages. If the attraction of the Archimedes can entice him to that machine, the results should be startling.



The blurb says that the Quest mouse package "is fully compatible with AMX Pagemaker, Super Art, the Max desktop, and Extra Extra as well as Wigmore Artist and both NovaCad and TechnoCad" and, of course, the digitized output from Watford Electronics' Video Digitizer.

As computer graphics becomes more interactive and user-friendly, the thickness of the graphics manual has become an insult, rather than a complement to the program. This one is A5 size, spiral bound, and despite the introduction stating that it was produced on a laser printer, the result looks distinctly like NLQ on a dot matrix printer. But on a package of this standard, we don't need 60 pages to convince us that we have value for money. Like most graphics manuals, it cries out for the odd diagram.

Of course, the clarity of diagrams would emphasise the thinness of content. Now that programs can use mostly icon-driven menus

and call on colour and animation where necessary, the manual might well have been reduced to a leaflet of technical preliminaries (few of us would read it, so it should fulfil the role just as well).

The mouse itself was a disappointment. The enthusiastic sales blurb describes it as 'chunky' and particularly suitable for children. In fact, it required my adult hand to maintain an uncomfortably steep angle, and this angle significantly reduced my control. This might have been fine for business use, but in art work, precision is the only advantage of the mouse over a Graf-pad style pen. As God has clearly ordained that brush, pen, pencil, scalpel and burin have the same shape the world over, why must the computer artist be condemned to shoulder-tip control? However, although the program requires a mouse interface, it accepts several breeds of mouse, so no criticism here.

The Quest Font Disc provides 29 (yes 29!) disc-based NLQ printer fonts. These are of the cheap and cheerful variety, and the absence of proportional spacing does nothing to improve their appearance. There is nothing of the quality or usefulness of a Univers or a good Roman, even allowing for the limitations of resolution. One is an un-acknowledged, low resolution copy of FUTURA BLACK, and one wonders about copyright. Other fonts are of dubious origin. The program can use the BBC's character set and the fonts from disc, with an option to tilt them (not italicise as the manual says).

For a further £7.50, an accompanying disc provides screens produced on the Quest Mouse package, and one of the best, by John Lavery is reproduced here.

In general, the package is a substantial advance over other packages of the same ilk on this and other micros. It compares favourably with some expensive packages on IBMs, and with the best packages I have seen on any other 8-bit micro. However it feels most competent when in use. It's 'friendly' for the general user rather than the enthusiast, and it demonstrates that home-user's graphics software has come of age. **B**

Personal Appointments Diary



In good time for the New Year, Geoff Bains has produced a computerised Appointments Diary. Just the way to get yourself organised for 1988.

Many diary programs written in Basic have been produced for the BBC micro (not least that in Vol.4 No.7), but this one provides several advantages. The diary can be created for any year after 1900, while the use of an index file means that both initial file creation, and subsequent file access are fast and efficient.

Type the program in and save to disc, preferably a blank one to provide enough space for the data files, although these can be on any disc. When you run the diary program, first enter the year, and the disc in the current drive is checked to see if the data files for that year already exist. If not, you can ask for new blank files to be created. Unlike most such programs, this only takes a few seconds. A calendar is then displayed for the January of that year, so that you can select whichever day you wish. The program provides displays of both one month at a time in calendar mode, and of individual days. The Tab key switches between the two display modes.

By using the left and right cursor keys, you can move forwards or backwards through the days, in either mode. If you hold down the Shift key at the same time, the display will move a month at a time (to the same date, or the nearest possible).

In the day display mode, the entries for that day are displayed in four sections, for the first and second parts of both the morning and afternoon. Each section can contain an ASCII entry of up to 50 characters.

December 1987

Su	Mo	Tu	We	Th	Fr	Sa
		1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30	31		

← → : date + Shift : month Tab : select

Month to view

To edit individual entries, you should press Copy whilst in day mode. A block cursor will then appear at the start of the first entry, and can be moved to the left or right by pressing the appropriate cursor keys. Text can then be typed in at any point (and you are always in insert mode). Pressing Delete works in the normal way, and Shift-Delete will erase the whole entry. By pressing the cursor up or down keys you can move between the four entries for the day. When you're satisfied with the entries, pressing Copy will save them to disc.

In edit mode the entries can also be 'marked' - labelled for future reference and search. There are five labels provided which can be used as required by the user to represent, say, birthdays, business meetings, doctor's appointments and the like. The entries are marked using the function keys f0 to f4. The keys work with a toggle action - repeated presses switch the labels on and off. All five labels can be set with one keypress using key f8, and all can be removed by pressing f9. It's recommended that you draw up a function key strip with the labels you require on the first five keys (just the job for the Keystrip Generator published in BEEBUG Vol.6 No.4).

The markers are used by the program to search through the diary entries for dates to remember. In day mode, pressing any of the function keys f0-f5 along with either the left or right cursor keys, searches through the entries, forwards or backwards, from the displayed date for any entry that has the respective marker set. If a suitably marked entry is found, the day is displayed. Otherwise the original day is returned. A search for entries marked with all

five markers can be made with f8, and for any entry with or without markers (but with some text) with f9, both in conjunction with either the left or right cursor keys to direct the search.

You should only exit from the Appointments Diary using the Escape key. Do not use Break! The Escape key saves the modified diary index (see below) to disc, and so preserves all the changes made to the diary entries since running the program. For brevity, the program does not include any protection against the data disc being removed or corrupted. The user must guard against this. Both data files and the program can happily sit on even a single sided 40 track disc, and to create a diary disc in this way is probably the best policy.

PROGRAM NOTES

Most other Basic diary programs suffer from a major failing - a lack of speed. This is mainly due to the speed of searching the disc for diary entries even when there are few or none there.

To solve this problem, the program uses an index file as well as the ASCII file of the entries themselves. This index has a byte reserved for each of the four entries for each day of the year. The index is loaded from disc at the start of the program and kept and updated in RAM. It is saved back to disc when Escape is pressed to end the session.

The index bytes are each made up of a series of flags to indicate the state of the entry to which they refer:

bit 7	set if any text in the entry
bit 6	not used
bit 5	not used
bit 4	set if marker f4 set
bit 3	set if marker f3 set
bit 2	set if marker f2 set
bit 1	set if marker f1 set
bit 0	set if marker f0 set

The index also allows faster generation of a new diary file. The ASCII file containing the diary entries can be (and initially is) filled with garbage. As long as the index file is zeroed, the garbage in the ASCII file won't be read or displayed. The ASCII file of the entries themselves comprises standard disc string variables (created with PRINT#), each containing the four field strings for each day.

```

10 REM Program Diary
20 REM Version B1.21
30 REM Author Geoff Bains
40 REM Beebug Jan/Feb 1988
50 REM Program subject to copyright
60 :
100 ON ERROR GOTO3160
110 MODE 7
120 L%=50
130 VDU23,0,10,64,0;0;0;
140 DIM entry$(3)
150 DIM INDEX% 1464
160 DIM start% 30
170 start%=&900
180 FOR pass=0 TO 2 STEP 2
190 P%=start%
200 [OPT pass:PHP:PHA:TXA:PHA:TYA:PHA:
LDA#151:LDX#32:LDY#203:JSR&FFFF4
210 .RET PLA:TAY:PLA:TAX:PLA:PLP:RTS:]
220 NEXTpass
230 ?&0220=start%MOD256: ?&0221=start%D
IV256
240 :
250 PROCgetyear
260 month%=1:date%=1
270 C%=OPENUP ("D"+STR$(year%))
280 REPEAT
290 PROCgetdate
300 PROCdisplay:PROCgetentries(day%)
310 PROCmove
320 UNTIL FALSE
330 END
340 :
1000 DEF PROCsearch(I%,J%)
1010 VDU28,3,24,39,5,12
1020 PRINT TAB(12,5)"Searching"
1030 O%=day%:K%=(leap%-366)*(I%>0)-1
1040 REPEAT:day%=day%+I%
1050 F%=FALSE:FOR entry%=0 TO 3
1060 IF (?(INDEX%+day%*4+entry%) AND J%
)=J% F%=TRUE
1070 NEXT
1080 UNTIL F% OR day%=K%
1090 IF day%=K% day%=0%
1100 day%=day%-I%
1110 ENDPROC
1120 :
1130 DEF PROCgetday
1140 RESTORE3110:day%=date%-1
1150 FOR I%=1 TO month%:READ monthtot%:
day%=day%+monthtot%
1160 NEXT I%

```


Tuesday 15th December 1987

Early am : Answer Technical Queries

3

Late am : Transfer articles to
Macintosh

0 2

Early pm : Planning meeting

Late pm :

+Shift:month f0=5,8,9:search
Copy:edit Tab:calendar

Day to view

```

1170 day%=day+leap%*(month%>2)-monthto
t%
1180 monthtot%=monthtot%+leap%*(month%=
2)
1190 IF date%>monthtot% day%=day%-date%
+monthtot%:date%=monthtot%
1200 ENDPROC
1210 :
1220 DEF PROCmove
1230 *FX21,0
1240 *FX4,1
1250 G%=GET:IF G%=9 ENDPROC
1260 IF G%<135 OR G%>137 GOTO1250
1270 IF G%=135 PROCedit:GOTO1380
1280 I%=(G%=136)-(G%=137)
1290 IF INKEY(-1) month%=month%+I%:mont
h%=month%+12*((month%>12)-(month%<1)):PR
OCgetday:GOTO1380
1300 IF INKEY-33 PROCsearch(I%,1)
1310 IF INKEY-114 PROCsearch(I%,2)
1320 IF INKEY-115 PROCsearch(I%,4)
1330 IF INKEY-116 PROCsearch(I%,8)
1340 IF INKEY-21 PROCsearch(I%,16)
1350 IF INKEY-119 PROCsearch(I%,31)
1360 IF INKEY-120 PROCsearch(I%,128)
1370 day%=day+I%
1380 day%=(day%+365-leap%)MOD(365-leap%
):PROCparams(day%):PROCdisplay:PROCgeten
tries(day%)
1390 GOTO1230
1400 ENDPROC
1410 :
1420 DEF PROCparams(day%)
1430 RESTORE3110:I%=0:month%=0
1440 REPEAT
1450 month%=month%+1:READ monthtot%:I%=
I%+monthtot%+leap%*(month%=2)

```

```

1460 UNTIL I%>day%
1470 date%=day%-I%+monthtot%+1+leap%*(m
onth%=2)
1480 RESTORE3100
1490 FOR I%=1 TO month%
1500 READ month%
1510 NEXT I%
1520 dayweek%=(day%+newyear%)MOD7
1530 RESTORE3090:FOR I%=0 TO dayweek%:R
EAD day$:NEXT
1540 ENDPROC
1550 :
1560 DEF PROCdisplay
1570 VDU28,3,24,39,0,12
1580 PRINT TAB(13,22)"[ ]:date'" +Shi
ft:month +f0=5,8,9:search'"SPC(6)"Copy
:edit Tab:calendar";
1590 FOR Y%=1 TO 2:PRINT TAB((23-(date%
>9)-LEN(month%+day%))/2,Y%)CHR$141;day%;
" ";date%:;IF date%MOD10=1 AND date%<11
PRINT;"st ";:GOTO1630
1600 IF date%=2 OR date%=22 PRINT;"nd "
;:GOTO1630
1610 IF date%=3 OR date%=23 PRINT;"rd "
;:GOTO1630
1620 PRINT;"th ";
1630 PRINT;month%;" ";year%:NEXT
1640 VDU28,2,20,39,6
1650 FOR Y%=0 TO 12STEP4:PRINTTAB(0,Y%)
CHR$129 TAB(0,Y%+2)CHR$130;TAB(11,Y%)CHR
$132:NEXT
1660 VDU28,3,24,36,6
1670 PRINTTAB(0,0)"Early am :";TAB(0,4)
"Late am :";TAB(0,8)"Early pm :";TAB(0,
12)"Late pm :";
1680 ENDPROC
1690 :
1700 DEF PROCgetdate
1710 VDU28,3,24,39,0,12,23,1,0;0;0;0;
1720 PRINT TAB(0,23)"[ ]:date +Shift:
month Tab:select"
1730 RESTORE3100
1740 FOR I%=1 TO month%
1750 READ month%
1760 NEXT I%
1770 FOR I%=1 TO 2:PRINTTAB((28-LENmont
h%)/2,I%)CHR$141;month%;" ";A$:NEXT
1780 VDU28,3,20,39,6:FOR I%=0 TO 1:PRIN
TTAB(3,I%)CHR$141;"Su Mo Tu We Th F
r Sa":NEXT
1790 PROCgetday
1800 W%=(8+(day%+newyear%)MOD7-(date%MO
D7))MOD7

```



```

1810 FOR I%=1 TO monthtot%
1820 K%=(W%+I%-1)MOD7:J%=(W%+I%-1)DIV7
1830 PRINT TAB(4+4*K%,3+2*J%);SPC(-(I%<
10));I%
1840 NEXT I%
1850 F%=0:*FX4,1
1860 *FX14,4
1870 VDU23,1,1,0;0;0;0;:REPEAT
1880 PRINT TAB(5+4*((W%+date%-1)MOD7),3
+2*((W%+date%-1)DIV7));
1890 *FX21,0
1900 G%=GET
1910 IF G%=9 F%=1:GOTO1960
1920 IF G%<136 GOTO1900
1930 IF INKEY-1 F%=1:month%=month%-(G%=
137)+(G%=136) ELSE date%=date%-(G%=137)+
(G%=136)
1940 IF date%<1 month%=month%-1:F%=1:da
te%=31 ELSE IF date%>monthtot% month%=mo
nth%+1:F%=1:date%=1
1950 IF month%<1 month%=12 ELSE IF mont
h%>12 month%=1
1960 UNTIL F%
1970 VDU23,1,0;0;0;0;
1980 *FX13,4
1990 *FX4,0
2000 IF G%<9 GOTO1710
2010 PROCgetday:PROCparams(day%)
2020 ENDPROC
2030 :
2040 DEF PROCgetentries(day%)
2050 FOR entry%=0 TO 3
2060 entry$(entry%)=STRING$(L%," ")
2070 NEXT entry%
2080 IF !(INDEX%+day%*4)=0 ENDPROC
2090 PTR#C%=(4*L%+2)*day%:INPUT#C%,text
$
2100 FOR entry%=0 TO 3
2110 IF?(INDEX%+day%*4+entry%)=0 GOTO2
180
2120 entry$(entry%)=MID$(text$,L%*entry
%+1,L%)
2130 VDU28,14,entry%*4+8,38,entry%*4+6:
PRINTentry$(entry%);
2140 VDU28,3,entry%*4+8,38,entry%*4+8
2150 FOR I%=0 TO 4
2160 IF((?(INDEX%+day%*4+entry%)) AND
(2^I%))>0 PRINT TAB(I%*2,0);I%;
2170 NEXT I%
2180 NEXT entry%
2190 ENDPROC
2200 :
2210 DEF PROCedit

```

```

2220 VDU28,3,24,39,22,12
2230 PRINTSPC4[" ]:cursor Up/Down:fiel
ld""SPC8"f0-5,8,9:mark field""Shift+Del
:delete field Copy:store";
2240 entry%=0
2250 VDU28,14,entry%*4+8,38,entry%*4+6
2260 *FX4,1
2270 *FX225,140
2280 VDU30,23,1,1,0;0;0;0;
2290 I%=1
2300 G%=GET
2310 VDU23,1,0;0;0;0;
2320 IF G%=127 AND INKEY(-1) entry$(ent
ry%)=STRING$(L%," ") :I%=1:VDU12,30:GOTO2
460
2330 IF G%=135 OR G%=138 OR G%=139 GOTO
2480
2340 IF G%<140 OR G%>149 OR (G%>144 AND
G%<148) GOTO2410
2350 VDU28,3,entry%*4+8,38,entry%*4+8
2360 IF G%=149 VDU12:?(INDEX%+day%*4+en
try%)=128:GOTO2390
2370 IF G%=148 PRINT"0 1 2 3 4";:?(INDE
X%+day%*4+entry%)=31:GOTO2390
2380 G%=G%-140:PRINTTAB(G%*2,0)::J%=? (I
NDEX%+day%*4+entry%):IF (J% AND (2^G%))<
>0 PRINT;" " :?(INDEX%+day%*4+entry%)=J%
-(2^G%) ELSE PRINT;G%;:?(INDEX%+day%*4+e
ntry%)=J% OR (2^G%)
2390 VDU30,28,14,entry%*4+8,38,entry%*4
+6
2400 IF I%>1 FOR J%=2 TO I%:VDU9:NEXT:G
OTO2460
2410 IF G%=136 AND I%>1 I%=I%-1:VDU8
2420 IF G%=137 AND I%<L% I%=I%+1:VDU9
2430 IF G%=127 AND I%>1 VDU8:I%=I%-1:PR
INT;RIGHT$(entry$(entry%),L%-I%);" " :en
try$(entry%)=LEFT$(entry$(entry%),I%-1)+
RIGHT$(entry$(entry%),L%-I%)+ " ":VDU30:I
F I%>1 FOR J%=2 TO I%:VDU9:NEXT
2440 IF G%<32 OR G%>126 GOTO2460
2450 IF I%=L%VDUG%,8:entry$(entry%)=LEF
T$(entry$(entry%),L%-1)+CHR$G%ELSEPRINT;
CHR$G%;RIGHT$(LEFT$(entry$(entry%),L%-1)
,L%-I%);:entry$(entry%)=LEFT$(entry$(ent
ry%),I%-1)+CHR$G%+RIGHT$(LEFT$(entry$(en
try%),L%-1),L%-I%):VDU30:I%=I%+1:FOR J%=
2 TO I%:VDU9:NEXT
2460 VDU23,1,1,0;0;0;0;
2470 GOTO2300
2480 *FX4
2490 *FX225,1
2500 VDU23,1,0;0;0;0;

```



```

2510 IF G%=138 entry%=entry%-(entry%<3)
:GOTO2250
2520 IF G%=139 entry%=entry%+(entry%>0)
:GOTO2250
2530 text$=entry$(0)+entry$(1)+entry$(2)
+entry$(3)
2540 PTR#C%=(4*L%+2)*day%
2550 PRINT#C%,text$
2560 !(INDEX%+day%*4)=!(INDEX%+day%*4)
OR &80808080
2570 FOR entry%=0 TO 3
2580 IF entry$(entry%)=STRING$(L%," ")
?(INDEX%+day%*4+entry%)=0
2590 NEXT
2600 ENDPROC
2610 :
2620 DEF FNfile(F$)
2630 C%=OPENUP(F$)
2640 CLOSE#C%
2650 =C%
2660 :
2670 DEF PROCosc(A$)
2680 $&980=A$
2690 X%=&80:Y%=&09:CALL&FFF7
2700 ENDPROC
2710 :
2720 DEF PROCgetyear
2730 VDU26,12,23,1,0;0;0;0;
2740 *FX4
2750 FOR I%=24 TO 0STEP-1:PRINTTAB(0,I%
)CHR$135;CHR$157;CHR$132;:NEXT
2760 PRINTTAB(2,1)CHR$129;TAB(2,2)CHR$1
29;TAB(2,22)CHR$129;TAB(2,23)CHR$129;TAB
(0,4)SPC3;TAB(0,21)" "
2770 FOR I%=1 TO 2:PRINTTAB(10,I%)CHR$1
41;"Appointments Diary":NEXT
2780 PRINTTAB(10,8)"Please enter the ye
ar"TAB(13,23)"After 1900 only":FOR Y%=12
TO 14:PRINTTAB(15,Y%)CHR$132;CHR$157;CH
R$135;SPC(6);CHR$135;CHR$157:NEXT
2790 VDU23,1,1,0;0;0;0;
2800 PRINTTAB(18,13);
2810 A$="":I%=4
2820 G%=GET
2830 IF G%=127 AND I%<4 I%=I%+1:A$=LEFT
$(A$,LEN(A$)-1):VDU8,32,8:GOTO2820
2840 IF G%<48 OR G%>57 GOTO2820
2850 PRINT CHR$G%,:A$=A$+CHR$G%:I%=I%-1
2860 IF I% GOTO2820
2870 VDU23,1,0;0;0;0;
2880 year%=EVAL(A$)
2890 IF year%<1900 PRINT TAB(18,13)SPC4
;TAB(12,23)CHR$136:GOTO2790

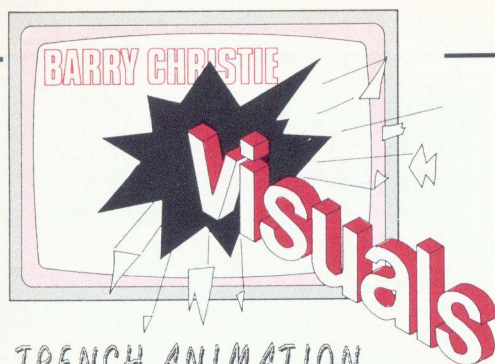
```

```

2900 VDU28,3,24,39,6,12,23,1,0;0;0;0;
2910 leap%=(year%=(INT(year%/4))*4)
2920 newyear%=(1.25*(year%-1900)-(NOTle
ap%))MOD7
2930 PROCparams(0)
2940 FOR I%=2 TO 3:PRINTTAB(14,I%)CHR$1
41;year%:NEXT
2950 IF leap% PRINTTAB(12,5)"Leap year
"
2960 PRINTTAB(6,9)"New years day: ";day
$
2970 ON ERROR GOTO3130
2980 IF FNfile("D"+STR$(year%)) AND FNF
ile("I"+STR$(year%)) PROCosc("LOAD I"+ST
R$(year%)+ " "+STR$~(INDEX%)):ENDPROC
2990 PRINTTAB(1,17)"No such year file.
New file? (Y/N)"
3000 G%=INSTR("YyNn",GET$)
3010 IF G%=0 GOTO3000 ELSE IF G%>2 GOTO
2730
3020 PRINTTAB(1,17)SPC(36)TAB(4,17)"Cre
ating file - Please Wait"
3030 PROCosc("SAVE D"+STR$(year%)+ " 0 +
"+STR$~(366*(4*L%+2)))
3040 FOR I%=INDEX% TO INDEX%+&5B4STEP4:
!I%=0:NEXT
3050 PROCosc("SAVE I"+STR$(year%)+ " "+S
TR$~(INDEX%)+ " +5B8")
3060 PRINTTAB(1,17)SPC(36)
3070 ENDPROC
3080 :
3090 DATA Sunday,Monday,Tuesday,Wednesd
ay,Thursday,Friday,Saturday
3100 DATA January,February,March,April,
May,June,July,August,September,October,N
ovember,December
3110 DATA 31,28,31,30,31,30,31,31,30,31
,30,31
3120 :
3130 ON ERROR OFF
3140 PROCosc("SAVE I"+STR$(year%)+ " "+S
TR$~(INDEX%)+ " +5B8")
3150 :
3160 ON ERROR OFF
3170 MODE 7
3180 *FX13,4
3190 *FX4
3200 *FX225,1
3210 CLOSE#0
3220 IF ERR<>17 REPORT:PRINT" at line
";ERL
3230 END

```

B



TRENCH ANIMATION

In this month's Visuals, Barry Christie makes use of machine code palette changes to implement fast smooth animation on a trench-like structure.

THE TRENCH

If you want to try out this animation, type in the program, and save it away. When you run it, you will first see a strange multi-coloured structure drawn on the screen. This takes quite a few seconds on a model B, but is much faster if you have a Master or a second processor. When the job is complete, the screen will crystallise into a trench-like structure, as in the accompanying picture. Then, as the caption says, use the cursor up and down keys. These will take you along the trench in either direction. The trench appears to move both rapidly and smoothly giving quite a lifelike appearance. The caption, on the other hand, does not move, and this could be replaced by the image of a creature or a vehicle.

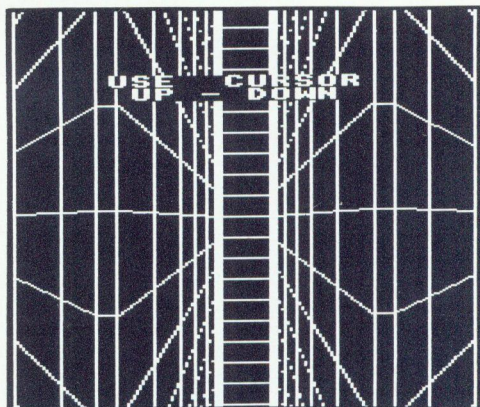
HOW IT WORKS

Essentially, this animation works by using palette changes on a mode 2 screen to display successive images of the trench, and so provide an impression of movement. It would not be possible to redraw the trench each time in a new position to simulate the movement, since a complete redraw would take several seconds. So in the procedure PROCbuildings, the trench image is drawn in 13 different positions, each position in a different logical colour (using colours 1 to 13). This explains the multi-coloured flashing "thing" which initially appears on screen. Finally, the non-moving structures on the trench are drawn in colour 14. This is set to green, and is not altered during the sequence.

The caption is drawn in colour 15, and is set to physical colour 7 (white). Again, this is not altered during the animation.

Next, PROCpalette is called. This sets each of the 13 logical colours 1 - 13 to physical colour zero, turning the whole trench black. If you reverse lines 100 and 110, you will obscure the initial drawing process from the user, but you would then need to display a "Please Wait" message to let him know what was going on.

Once the trench has been drawn and blanked out, the machine code is assembled and called. This first checks if the up or down cursor keys have been pressed. It achieves this by reading the contents of the zero page location &EC, which contains the value of the last key pressed. The key number to be found in this location is the so-called internal key number of the pressed key +128. The internal key number, it should be noted is not the same as the negative INKEY value. See the Advanced User Guide for further details.



If either of the two vertical cursor keys have been pressed, the routine makes two OSBYTE calls to alter the palette accordingly, cycling through the colours, and making each logical colour in sequence appear as colour 2, or green. The palette call is made twice each time around the loop (at line 1550 and 1570), once to extinguish the previous image, and once to illuminate the next. Between the two calls OSBYTE 19 is used to synchronise the display updating with the monitor's vertical scan, thus minimising flicker.

The program alters the palette by calling OSBYTE &9B (115 decimal). As usual, the X register holds the value to be written. In this case, the byte consists of two 4 bit fields. The top 4 bits contains the logical colour and the bottom 4 bits the corresponding physical colour. Thus, calling OSBYTE &9B with &72 in the X register will set logical colour 7 to physical colour 2. In mode 2 this will change normally white text and graphics to green. You can try this out by executing:

```
*FX 155,114
```

from mode 2. The value written, 114, is &72.

This is the exact equivalent of the VDU19 call:

```
VDU 19,7,2;0;
```

```
10 REM Program Trench Animation
20 REM Version B 0.5
30 REM Author Barry Christie
40 REM BEEBUG December 1987
50 REM Program subject to copyright
60 :
70 ON ERROR GOTO 1680
80 MODE 2
90 PROCinitialise
100 PROCbuildings
110 PROCpalette
120 COLOUR 15
130 PRINTTAB(4,5)"USE  CURSOR"
140 PRINTTAB(5,6)"UP - DOWN"
150 PROCassemble
160 CALL animate
170 END
180 :
1000 DEF PROCbuildings
1010 FOR I%=0 TO 1024 STEP step%
1020 c%=(c% MOD 13)+1:GCOL 0,c%
1030 s%=s%-2*step%
1040 MOVE 0,I%-s%:DRAW 226,I%-2*s%
1050 DRAW 290,I%-2*s%:DRAW 575,I%
1060 DRAW 705,I%:DRAW 990,I%-2*s%
1070 DRAW 1054,I%-2*s%:DRAW 1280,I%-s%
1080 NEXT
1090 GCOL 0,14
1100 FOR I%=1 TO 9
1110 MOVE hor1%-s,0:DRAW hor1%-s,1024
1120 MOVE hor2%+s,0:DRAW hor2%+s,1024
1130 s=s+(s/2)
1140 NEXT
1150 MOVE 0,0:DRAW 0,1023:MOVE 1279,0
1160 DRAW 1279,1023
1170 MOVE 226,0:DRAW 226,1023
1180 MOVE 1054,0:DRAW 1054,1023
1190 ENDPROC
```

```
1200 :
1210 DEFPROCpalette
1220 FOR A=1 TO 13
1230 VDU19,A,0;0;
1240 NEXT
1250 ENDPROC
1260 :
1270 DEF PROCinitialise
1280 VDU 23,1,0;0;0;0;
1290 VDU 19,14,2;0;
1300 VDU 19,15,7;0;
1310 step%=4:*FX4,1
1320 s%=1024:c%=7:s=18
1330 hor1%=575+s:hor2%=704-s
1340 ENDPROC
1350 :
1360 DEF PROCassemble
1370 ?&70=&A9:&?&71=&B9 :REM Key
numbers
1380 FOR pass%=0 TO 2 STEP 2
1390 P%=&900
1400 [ OPT pass%
1410 .animate
1420 LDY #1
1430 .loop
1440 LDA &EC:CMP &70:BNE press1:INY
1450 .press1
1460 LDA &EC:CMP &71:BNE press2:DEY
1470 .press2
1480 CPY &#0E:BNE reset1:LDY &#01
1490 .reset1
1500 CPY &#00:BNE reset2:LDY &#0D
1510 .reset2
1520 TYA:STY &73
1530 ASL A:ASL A:ASL A:ASL A
1540 STA &72:ORA &#02:TAX
1550 LDA &#9B:JSR &FFF4 \Write palette
1560 LDA &#13:JSR &FFF4 \Wait for sync
1570 LDA &#9B:LDX &72:
1580 JSR &FFF4 \Write palette
1590 LDY &73
1600 BIT &FF:BMI escape
1610 JMP loop
1620 .escape \Escape
pressed
1630 RTS
1640 ]
1650 NEXT pass%
1660 ENDPROC
1670 :
1680 *FX4
1690 MODE 7
1700 REPORT:PRINT" at line ";ERL
1710 END
```

B

LOAD AND GO WITH BASIC PROGRAMS

*Bernard Hill's short utility will parcel up your Basic programs so that they can be *RUN just like machine code.*

How many times have you typed CHAIN? It must be thousands! It's particularly annoying when machine code programs don't even have to be *RUN. You can just type *<name> and in they come from disc and off they go. Now, that's all over. With this very short utility your Basic programs can be loaded just like machine code utilities. For the first time you can use that *OPT 4,2 to *RUN a Basic program called !BOOT. This utility is worth its weight in gold too in the ADFS. Now you can put Basic programs in the Library directory, and call them up from any directory just like machine code (the ADFS, for example, doesn't search the library directory when you use CHAIN).

To use it, simply type in the program and save (as usual with assembler programs) before running (I suggest the name BASAVE). When you run it, nothing happens except that the code is assembled and then saved as a utility BSAVE on the current directory. ADFS and other regular users of libraries can copy this to the library directory if they wish. Next time you want to SAVE a Basic program, instead of typing, for example, SAVE "fred", issue the command *BSAVE fred, and the program will be saved to disc in a form which allows ANY of the commands:

```
CHAIN "fred"
*RUN fred
*/fred
*fred
```

to start your program running. You can even LOAD "fred" and you won't be aware that it's any different from the normal form of Basic program using the regular SAVE command.

There is just one word of warning. Using *BSAVE fred will save 'fred' so that *fred reloads it back at the value of PAGE from which you saved it. If you're a regular PAGE changer (for instance to keep two programs at

once in memory), then you will need to take care. Of course, using CHAIN or LOAD on the file 'fred' will not give this problem.

```
10 REM Program Bsave
20 REM Version B0.9
30 REM Author Bernard Hill
40 REM Beebug December 1987
50 REM Program subject to copyright
60 :
100 MODE 7:DIM os 40
110 FOR pass%=0 TO 3 STEP 3
120 P%=&A00:N%=P%
130 [OPT pass%
140 LDX #&A8:LDY #0:LDA #1
150 JSR &FFDA:LDA &A8:STA pb
160 LDA &A9:STA pb+1
170 .loop
180 LDA exec,Y:STA (2),Y:INY
190 CMP #96:BNE loop
200 TYA:TAX:LDY #1 : LDA &18
210 STA (2),Y:STA pb+11:STA pb+3
220 LDA 2:STA pb+6
230 LDA 3:STA pb+7
240 TXA:SEC:ADC 2:STA pb+14
250 LDA 3:ADC #0:STA pb+15
260 LDX #pb MOD 256:LDY #pb DIV 256
270 LDA #0:JSR &FFDD:RTS
280 .pb
290 BRK:BRK
300 BRK:BRK:BRK:BRK \ reload
310 BRK:BRK:BRK:BRK \ exec
320 BRK:BRK:BRK:BRK \ start
330 BRK:BRK:BRK:BRK \ end
340 .exec
350 LDA #0:STA &18
360 LDX #15:LDX #1:JSR &FFFF4
370 LDA #138:LDX #0
380 LDY #ASC"O":JSR &FFFF4
390 LDY #ASC"L":JSR &FFFF4
400 LDY #ASC"D":JSR &FFFF4
410 LDY #13:JSR &FFFF4
420 LDY #ASC"R":JSR &FFFF4
430 LDY #ASC"U":JSR &FFFF4
440 LDY #ASC"N":JSR &FFFF4
450 LDY #13:JSR &FFFF4
460 RTS
470 ]
480 NEXT
490 PROCoscli("SAVE Bsave "+STR$~N%+"
"+STR$~P%)
500 END
510 :
1000 DEF PROCoscli($os)
1010 X%=os MOD 256:Y%=os DIV 256:CALL &
FFF7
1020 ENDPROC
```

B

PRINTER SURVEY

(Part 2)

Geoff Bains continues to lead us through the printer jungle in the second part of his survey, which includes two extra printers for good measure.

SAMLECO DX-86, £276

Unusually, the Samleco DX-86 is made in the UK. It has a bulky, 'industrial' look with minimum effort devoted to the aesthetic design. The top of the machine hinges off to give access to every part of the innards. This machine has no paper bail. Instead a curved section of the plastic cover keeps the paper pressed around the roller. This arrangement is convenient but it can give problems with thick paper or multiple copies.

```
NLQ mode: ABCDEFGHIJK
LMNOPQRSTUVWXYZabcde f
ghijklmnopqrstuvwxyz0
123456789!"£$%&'()?/@
DRAFT mode: ABCDEFGHI
JKLMNOPQRSTUVWXYZabcd
efghijklmnopqrstuvwxyz
z0123456789!"£$%&'()?
```

However, the DX-86 does not produce very good print. The draft mode is very dotty and a little faint. The NLQ mode print is also poor compared to even the Amstrad DMP-2000 costing £80 less. The DX-86 is not that fast either. The claimed speed figures are high enough but the actual speeds are different - only 17cps in NLQ mode.

If the DX-86 was the cheapest printer around, its performance would be excusable but next to the quality of the Canon or the price of the Amstrad DMP-3000 it cannot compete.

*SAMLECO, Fairacre Industrial Estate,
Dedworth Road, Windsor, Berks SL4 4LE*

PANASONIC KX-P1091, £328

The KX-P1091 is a compact machine which produces surprisingly good quality print given its small NLQ dot matrix (only 12 x 18). It's not up to the standard of the Canon or Seikosha but is very reasonable. The printer is compatible with both IBM and Epson standards, but the graphics characters of the IBM character set cannot be printed in true NLQ. Instead, they are automatically printed as double strike characters which suffices quite well.

```
NLQ mode: ABCDEFGHIJK
LMNOPQRSTUVWXYZabcde f
ghijklmnopqrstuvwxyz0
123456789!"£$%&'()?/@
DRAFT mode: ABCDEFGHI
JKLMNOPQRSTUVWXYZabcd
efghijklmnopqrstuvwxyz
z0123456789!"£$%&'()?
```

There is a switch on the front of the machine to select between draft, NLQ and condensed printing. However, this cannot be over-ridden by control codes from the Beeb. Rather strangely, proportional spacing is not possible in NLQ mode. Like the Seikosha and Amstrad DMP-4000, the KX-P1091's tractor feed pulls the paper through the printer.

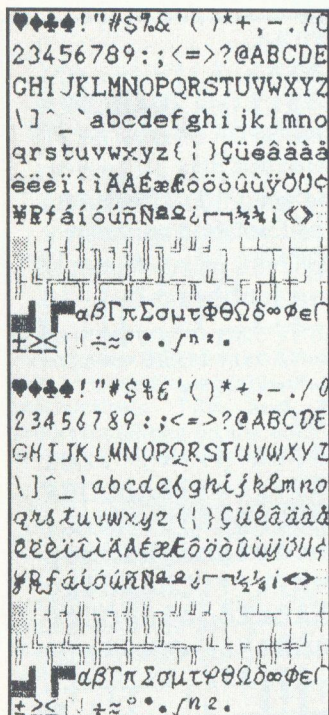
The KX-P1091 is a fairly standard NLQ printer with little to distinguish it from the crowd. For this price it cannot really compete with the Amstrad DMP-4000 or the Canon.

*PANASONIC, 280-290 Bath Road,
Slough SL1 6JG*

DICONIX 150, £459

The Diconix is extremely small. It is only the size of a large hardback novel and it weighs under 4lbs. This machine is battery-powered so it is genuinely portable. Although this feature will not interest most BEEBUG readers, it can be useful when the printer is to be moved between several Beebs. A clever feature is the fitting of the batteries inside the paper roller, saving considerably on the unit's size. Four U2 size rechargeable batteries are used and fully

charged they last for about 150 pages. A mains adaptor/charger is included for using the printer without batteries.



The Diconix is an ink jet printer and so it's very quiet. It can produce both draft and NLQ print, and has friction feed and tractor mechanisms.

Diconix claims this printer can produce draft text at 150cps but the average speed it can manage is only 73cps. In NLQ mode the claimed speed is 50cps but 30cps is a more realistic figure.

In NLQ mode the Diconix can print in two styles - the usual typewriter look-alike and a script font. Neither style is very good quality and they really require double strike printing for good effect. However, this slows the machine down and uses up the £10 ink cartridges rather quickly. The Diconix is both Epson and IBM compatible. This means it can produce the graphics symbols and foreign accented letters used by the IBM PC. Unusually, the Diconix has a special 'envelope' mode to print sideways on fanfold stationery envelopes.

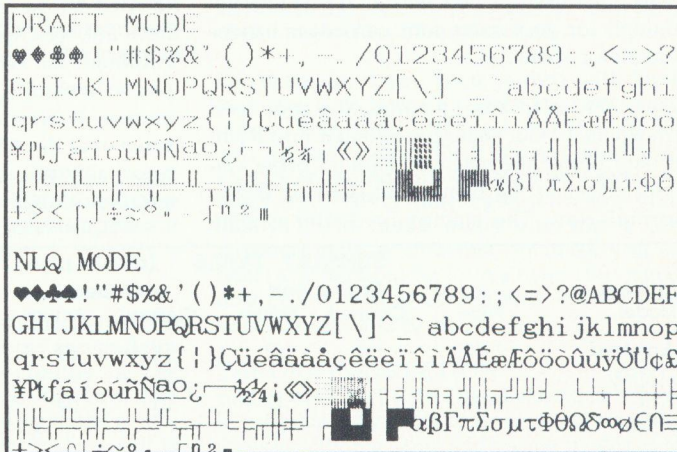
There is no other printer to approach the Diconix 150 for size and weight but, despite its two NLQ styles and sideways printing mode, you must have a real need for its portability to justify its relatively high price.

*DICONIX 150, c/o Norbain Technology,
14-16 Boulton Road, Reading, Berks RG2 0LT*

AMSTRAD LQ3500 £401

Of course Amstrad had to be first with a really cheap 24 pin dot matrix printer. The LQ3500 is in fact the cheapest 24 pin machine around. Whereas nine pin printers require two passes of the printhead over the paper, with a shift of the paper inbetween, to produce the overlapping dots which make up the quality of NLQ characters, 24 pin printers use two staggered rows of pins in the printhead, giving NLQ with just one pass. This makes 24 pin printers much faster in NLQ mode and (because they have 30% more dots per character) the quality is likely to be even better than nine pin NLQ.

The LQ3500 does indeed produce very reasonable quality print at a speed which compares well with similarly priced nine pin machines. However, the quality is not really up to expectations. Compared to other 24 pin printers (admittedly most cost twice the price or more) the LQ3500 cannot really hold its own. It is not as good as the best of the nine pin machines either (such as the Star NL-10).



What is more worrying is that the LQ3500 is built so flimsily too. It has none of the advanced features to be found on the other 24 pin printers and the up-market nine pin machines - it has no automatic paper loading, no RS232 interface or sheet feeder options. The LQ3500 is (naturally) IBM compatible, but has little to recommend it to Beeb users. Amstrad has been the first with a really cheap 24 pin printer but I think they have missed the mark.

AMSTRAD, 169 Kings Road, Brentwood, Essex CM14 4EF

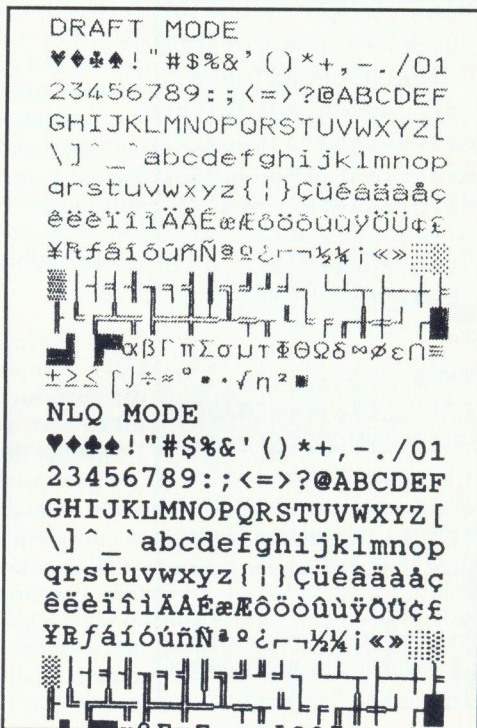
NEC P2200 £454

For just £50 more than the LQ3500, NEC's latest offering is also a 24 pin printer, this time with all the features and quality you can expect. The P2200 can print at a good speed, with a quality of NLQ that stands up to the best, and has many advanced features as well. Most peculiar is the replacement of the normal DIP switches with a non-volatile memory system set by an interactive menu - the printer prints the prompts on paper and you reply with presses of the front panel buttons for 'yes' or 'no'. It's weird but it works well.

The P2200 is Epson and IBM compatible, it can produce all the combinations of printing effects you could reasonably want and it can print in several fonts. Never mind the usual solitary typewriter look-alike you get with other NLQ printers. This one can manage that font and three other useful styles as well. If that lot's not enough for you, extra font cartridges can be bought for about £35.

The P2200 also has an unusual paper feed arrangement. Cut sheet paper is fed into a slot at the front of the machine. Fanfold paper can enter either at the front or at the rear in the normal way. The tractor feed unit can be

arranged to equally push or pull paper through the printer. Everyone should be satisfied with one of these arrangements.



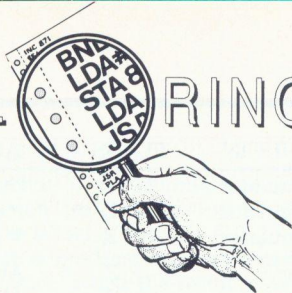
The P2200 is also built very sturdily. It is unusual to see a budget printer from NEC, and the normal quality of design and manufacture have not been spared from this cheaper model. Although the NEC is not the cheapest 24 pin printer around it is certainly the best value for money. It offers far more than £50 of extra value over the Amstrad LQ3500 and provides a reasonably priced entry into the future of dot matrix printing.

NEC Business Systems, 35 Oval Road, London NW1 7EA

SUMMARY TABLE (continued)

Make & Model	Price	Draft speed		NLQ speed		NLQ Quality	Supplier
		Claimed	Actual	Claimed	Actual		
Amstrad LQ3500	£401	135	62	64	31	7	Amstrad (0277) 228888
NEC P2200	£454	140	47	82	34	9	NEC 01-993 8111

B



ASSEMBLER

Part 6

A series for complete beginners to machine code by Lee Calcraft

This month: Using the Logic Instructions

AND ORA EOR TSB TRB BIT

We are going to take a look at six logic instructions this month. As we shall see, only four of these are available on the model B. The instructions TRB and TSB are new implementations on the 65C12 processor, and are only available on the Master and Compact.

The six instructions fall naturally into three groups depending upon the way in which the result of the instruction is stored. AND, ORA and EOR all place their result in the accumulator. TSB and TRB place theirs in memory, while BIT affects the flags only, leaving both accumulator and memory unchanged.

THE AND INSTRUCTION

The AND instruction behaves very similarly to its Basic counterpart, the operator **AND**. Both make a bit-by-bit comparison of their two operands, giving a result of one for each bit position in which the bits of both operands are one, and zero where they are not. An example should make this clearer:

	10101010	&AA
AND	11110000	&F0
	10100000	&A0

Here the binary numbers 10101010 and 11110000 constitute the two operands, and give a result of 10100000. You can test the operation in Basic quite simply by typing:

```
PRINT ~(&AA AND &F0)
```

In a similar way, the 6502 instruction **AND** logically ANDs the contents of the accumulator with memory, placing the result in the accumulator, and setting the Z (or Zero) flag and N (or Negative) flag according to the result, in the usual way. That is to say, if the result is

zero, the Z flag is set, if not it is unset; while the N flag is altered to reflect the state of the top bit of the result in the accumulator. If the top bit is set, then the N flag is set, otherwise it is unset.

The AND instruction can use a wide variety of addressing modes, as indicated in the table. For example, using immediate mode:

```
AND #&F0
```

will AND the contents of the accumulator with the hex byte &F0, while, in absolute mode,

```
AND &3000
```

will AND the contents of the accumulator with the contents of memory location &3000. More esoterically,

```
AND (&70),Y
```

will AND the accumulator with the contents of the location found by adding the contents of the Y register to the address derived from the contents of locations &70 and &71. This is the so-called indirect indexed addressing mode, treated in Part 4 of this series.

There are many uses for the AND instruction, and a closer look at the Basic example above may give a clue to a particularly important application. The assembler equivalent of the example would be something like:

```
LDA #&F0
AND #&AA
```

The result, &A0, will appear in the accumulator. From the binary representation of this operation, as given above, you can see that the byte 11110000 (&F0) has acted as mask on the other operand, 10101010 (&AA), masking out the bottom half of the byte to give the result 10100000 (or &A0). This can be extremely useful in a variety of applications.

For example, if you need to deal with inputs from the user port on certain bits, but not on others, you could just mask out the bits that you do not require. Or if for any reason you want just the bottom 4 bits of any byte, you need only AND that byte with the value 00001111 to mask away the top 4 bits. Alternatively, with a one-bit mask you can select any required bit from a byte. Thus a mask of 00000001 will select bit zero, 00000010 will select bit one, and so on.

THE ORA INSTRUCTION

Just as with AND, the ORA instruction behaves in a similar way to its Basic counterpart, **OR**. It takes exactly the same addressing modes as the

	MEANING	RESULT IN	FLAGS	ADDRESSING
AND	AND memory with accumulator	Accumulator	N Z	** see below
EOR	Exclusive OR memory with acc	Accumulator	N Z	** see below
ORA	OR memory with accumulator	Accumulator	N Z	** see below
TRB*	AND comp of acc with mem	Memory	Z	Abs + Zero page
TSB*	OR accumulator with memory	Memory	Z	Abs + Zero page
BIT	Test bits in memory with acc	Flags only	NVZ	Abs + Zero page
Key	* Only available on Master series machines. ** The following addressing modes are supported: Immediate Zero page indirect* Indirect indexed Absolute Absolute indexed Pre-indexed indirect Zero page Zero page indexed			

Table showing 6502/65C12 logical instructions

AND instruction, and affects the same two flags Z and N in the same way. Again the effect of the operation is best understood by example:

```

10101010    &AA
OR  11110000    &F0
-----
11111010    &FA

```

As you can see from the result, the ORA instruction makes a bit-by-bit comparison of its two operands, placing a one in the result in any field where there is a one at the corresponding bit position in either or both of the operands, otherwise it sets the bit-result to zero. Again you can experiment with the OR operator from Basic. For example, try:

```
PRINT ~(&AA OR &F0)
```

The equivalent of this in assembler is:

```

LDA #&F0
ORA #&AA

```

The most obvious use for the ORA instruction is to set selected bits of a byte. Thus, if you OR the binary number 10000000 with any byte, it will set the top bit of that byte, without affecting any other bit-field. By contrast, AND can be used to *unset* a bit as follows. If you AND any byte with binary 01111111, it will *unset* the top bit, leaving all other bits unchanged. Of course, the two 6502 instructions AND and ORA do not write back the new value to the memory location; the result is left in the accumulator, and this is where the two new Master-series-only instructions TRB and TSB come into their own, as we shall see in a moment.

THE EOR INSTRUCTION

EOR is the last instruction of the group. Again it behaves just like its Basic counterpart, the operator EOR, and uses the same addressing

modes, and affects the same two flags (Z and N), as its two bedfellows. Here is an example:

```

10101010    &AA
EOR 11110000    &F0
-----
01011010    &5A

```

From the result you can probably see what is going on. The EOR instruction works similarly to the ORA, except that where the corresponding bits in both operands are one, it gives a zero bit-result. To test this in Basic, try:

```
PRINT ~(&AA EOR &F0)
```

There are a number of uses for Exclusive OR operations in assembler. Many are based on a property discernible from the example. In the top 4 bits of the operation, the result is the complement of the first operand, while in the lower 4 bits, where the second operand is zero, the result is a copy of the first operand. Thus if we perform an Exclusive OR with the value &FF, we will produce the complement of the other operand. This can often be useful.

Another interesting feature of the Exclusive OR operator, is that repeating an Exclusive OR operation reproduces the original operand. Thus, while:

```

&AA EOR &F0 gives &5A
&5A EOR &F0 gives &AA

```

and by the same token,

```
&5A EOR &AA gives &F0
```

Another property of the EOR instruction is that it can be used to compare two bytes, since only when two operands are exactly the same will the result of an Exclusive OR be zero. The result can then be easily tested by checking the zero flag. Of course, the CMP instruction is normally used for comparing bytes, and it also signals a match by setting the zero flag.

THE TSB INSTRUCTION

This new instruction is very similar indeed to the ORA instruction dealt with above. Its effect is to perform a bitwise logical OR of the accumulator with a specified memory location. The major difference is that the result appears, not in the accumulator, but in memory. This allows the operation to be used to set specified bits at any memory location. Indeed, it is from this particular application that the instruction's mnemonic is derived: **TSB - Test and Set Bits**. It *tests* by setting the zero flag in the unusual manner outlined below.

To see how TSB works, consider the following:

```
LDA #&80
TSB &3000
```

The first instruction loads the accumulator with &80 (or 10000000 binary). The second will set the top bit (i.e. as specified by the 10000000 in the accumulator) of the contents of location &3000, without affecting its other bit-fields. The zero flag will be set if the result of performing an **AND** instruction (not **ORA**) on the accumulator and the memory location *would have been* zero. This is much more use than setting Z according to the *actual* result of the TSB operation, since the way in which it works results in the Z flag reflecting the state of the bit (or bits) in memory which are to be reset by the instruction. In the case of our example, Z=1 if the top bit of &3000 was zero prior to the execution of TSB, and 0 if it was not.

THE TRB INSTRUCTION

The function of the TRB instruction is the reverse of TSB. Its mnemonic is again derived from its anticipated application. It is used to *Test and Reset Bits*. Thus the following two instructions:

```
LDA #&80
TRB &3000
```

would simply reset (or set to zero) the top bit of memory location &3000, leaving the other seven bits unaltered. Again, this is achieved by using a mask of &80, or 10000000 binary. TSB, remember, would *set* the top bit under the same circumstances. And just as with TSB, the zero flag is set if the result of performing an **AND** operation between the accumulator and memory *would have been* zero. In other words, Z is again set to reflect the state of the bit (or bits) in memory which are to be altered by the TRB instruction.

Executing the instruction TRB when the accumulator holds the value 10000000 binary is very similar in effect to loading the accumulator with 01111111, executing an **AND** instruction with memory, and storing the result back in memory. Indeed, this is how users of the 6502 (as opposed to the 65C12) might reset the top bit of a memory location.

THE BIT INSTRUCTION

Lastly, we come to the BIT instruction. This is available on all versions of the 6502 processor, and is the only one of our six logical instructions which does not generate a direct result. In this respect it is similar to the **CMP** instruction, which we will be looking at in the next issue. Both **BIT** and **CMP** alter the state of the processor's flags, but neither of them change any other register or memory location.

The mnemonic for the BIT instruction derives from *test bits in memory with accumulator*. In practice it is used with the accumulator holding a mask for the bits to be tested. Thus if we execute the following instructions:

```
LDA #&80
BIT &3000
```

the zero flag will be set if the top bit (i.e. as defined by the 10000000 in the accumulator) in location &3000 contains a zero, otherwise it will be unset. In practice what the BIT instruction does is to perform a simulated **AND** operation with the accumulator and the memory as operands, but the result is not recorded. Both operands are left unchanged, and the zero flag is set according to the result of the simulation.

If the operand supplied with the instruction is in memory, and not an immediate operand (i.e. not a constant), then two other flags are also altered as a result of the BIT instruction: the N and the V flags. These are set according to the *original value* held in memory. Unusually, they are not used to reflect the *result* of the BIT operation. Bit 7 of the operand is copied into the Negative flag, and bit 6 is copied into the Overflow flag (V). This enables the BIT instruction to get information from any memory location without overwriting the contents of the accumulator or the X and Y registers. This can be useful on occasion, as you may appreciate, but the most common use of the BIT instruction is in reading the value of a particular bit-field of a byte held in memory. There is, however, an important limitation to

the use of the BIT instruction. Its addressing modes, as you can see from the table, are very limited, and because it cannot be used with indexed addressing of any type, it is no use for testing sets of data held in a table. For this purpose, the much more flexible CMP instruction must be used.

LOGIC SIMULATOR

To round off this month's article, I have included a short program which simulates the effects of the three instructions AND, OR and EOR. When it is run, it requests two binary numbers (8 bits each). After supplying these, the user should enter an "A" an "E" or an "O" to select the required logical instruction. The program then prints the result in binary and hex, and indicates the resultant state of both the Zero and Negative flags. It would be a relatively easy matter to extend the program to cover the other three logical instructions dealt with this month. The program, I should stress, is purely a simulation, and is written entirely in Basic.

```

10 REM Assem Ver. B1.6
20 REM Logic Demonstration
30 REM By Lee Calcraft
40 :
50 REPEAT:MODE4
60 PROCtext
70 acc=FNget(18,10)
80 mem=FNget(18,12)
90 ans=EVAL(STR$acc+FNop(18,14)+STR$m
em)
100 PROCput(ans,18,18)
110 PROCflags
120 PRINTTAB(2,31)"Space to continue o
r
Q to quit ";
130 UNTIL GET<>32:PRINT:END
140 :
150 DEFFNget(X,Y)
160 N=7:Z=0:PRINTTAB(X,Y);
170 REPEAT
180 REPEAT:A=INKEY(0)
190 UNTIL A=48 OR A=49
200 Z=Z+(A-48)*2^N
210 N=N-1:PRINTCHR$(A);
220 UNTIL N=-1
230 Z$=STR$~Z:IF Z<16 THEN Z$="0"+Z$
240 PRINTSPC(5)"&";Z$
250 =Z
260 :
270 DEFFNop(X,Y)

```

LOGIC DEMONSTRATION

Accumulator	10101010	8AA
Memory	11110000	8F0
Instruction	AND	
Result	10100000	8A0
Flags	Z = 0 N = 1	

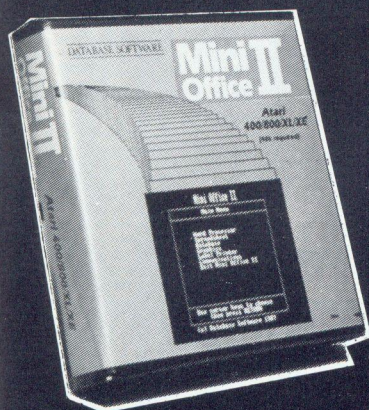
Enter binary data
Then A (AND) E (EOR) or O (OR)
Space to continue or Q to quit

```

280 PRINTTAB(X,Y);
290 REPEAT:A$=GET$:UNTIL A$="A" OR A$=
"E" OR A$="O"
300 IF A$="A" THEN op$="AND"
310 IF A$="E" THEN op$="EOR"
320 IF A$="O" THEN op$="OR"
330 PRINTop$;IF op$="OR" PRINT"A"
340 =op$
350 :
360 DEFPROCput(ans,X,Y)
370 N$="":PRINTTAB(X,Y);
380 FOR N=7 TO 0 STEP -1
390 bit=2^N
400 PRINTSTR$-((bit AND ans)>0);
410 NEXT
420 ans$=STR$~ans:IFans<16 ans$="0"+an
s$
430 PRINTSPC(5)"&";ans$
440 ENDPROC
450 :
460 DEFPROCflags
470 PRINTTAB(18,20)"Z = ";-(ans=0)
480 PRINTTAB(18)"N = ";-((ans AND 128)
>0)
490 ENDPROC
500 :
510 DEFPROCtext
520 PRINTTAB(15,1)"L O G I C"
530 PRINTTAB(13,3)"DEMONSTRATION"
540 PRINTTAB(2,10)"Accumulator"
550 PRINTTAB(2,12)"Memory"
560 PRINTTAB(2,14)"Instruction"
570 PRINTTAB(2,18)"Result"
580 PRINTTAB(2,20)"Flags"
590 PRINTTAB(2,27)"Enter binary data"
600 PRINTTAB(2,29)"Then A (AND) E (EOR)
or O (OR)"
610 ENDPROC

```


Mini Office II



Mini Office II is a comprehensive integrated suite of programs setting a new standard in home and business software.

Most of the wide range of features – many of which are usually only available on software costing hundreds of pounds – are easily accessed by using cursor keys to move up and down a list of options and pressing Return to select.

Mini Office II takes over where the original, highly successful Mini Office left off, with numerous extra features, two additional modules, a program to convert existing Mini Office files to Mini Office II format, and a 60 page, easy to follow manual.

And it's at a price everyone can afford!

B, B+	Tape £14.95
Master	5.25" disc £16.95
Master Compact	5.25" disc £19.95
	3.5" disc £21.95

B, B+, Master,	4 by 32k Rom
Master Compact	board £59.95

6 powerful home and business programs in just one package – at a price that simply can't be matched!

- **WORD PROCESSOR:** Compose a letter, set the print-out options using embedded commands or menus, use the mail merge facility to produce personalised circulars – and more!
- **DATABASE:** Build up a versatile card index, use the flexible print out routine, do powerful multi-field sorting, perform all arithmetic functions, link with the word processor – and more!
- **LABEL PRINTER:** Design the layout of a label with the easy-to-use editor, select label size and sheet format, read in database files, print out in any quantity – and more!
- **SPREADSHEET:** Prepare budgets or tables, total columns or rows with ease, copy formulae absolutely or relatively, view in either 40 or 80 column modes, recalculate automatically – and more!
- **GRAPHICS:** Enter data directly or load data from the spreadsheet, produce pie charts, display bar charts side by side or stacked, overlay line graphs – and more!
- **COMMS MODULE:** Using a modem you can access services such as MicroLink and book rail or theatre tickets, send electronic mail, telex and telemessages in a flash – and more!

INSTANT Mini Office II

Now available on a power-packed rom board!

- Four 32k roms on one board packed with 128k of super fast machine code.
- Easy to install – no soldering needed.
- Split second application selection.
- Frees valuable ram space for much more data.
- Fully compatible with the whole BBC Micro range, Aries or Watford shadow ram board, and tape, DFS, ADFS and ECONET filing systems.

The full selection of Mini Office II is available through BEEBUG Retail at MEMBER'S DISCOUNT. Please refer to your BEEBUG Retail Catalogue

DATABASE SOFTWARE

Europa House, 68 Chester Road,
Hazel Grove, Stockport SK7 5NY

Personal Ads

Morley Teletext adaptor, ATS and support ROMs, two handbooks, and keystrip £80. Tel. (0920) 5406.

Master 128 complete with Master ROM, Commstar 2, 2 cartridges, Reference manuals 1 & 2, and Digimouse £400. Opus 800K twin disc drive & 20 discs £150. Microvitec colour monitor 1431 £100. Or, £600 the lot, including some other books and programs. Tel. (0205) 68276 after 6pm.

Master 128 fitted with turbo co-processor and twin Watford 40/80 CD800S drives. All still under guarantee. C/W Interword, Wordwise II, Microprolog, Commstar II, Masterfile II, Whiteknight and about 50 discs of software and all manuals. £850 ono. Tel. (0962) 713546.

For sale: Studio 8 Disc £10, Castle Quest disc £6, Exploding Fist disc £6, Sentinel disc £6. Contact Ben at 8c Spencer Hill, London SW19 4NY.

AMX Mouse with Super Art £30, Viglen 100K Single disc drive £30, 6502 Second Processor £50, Acorn 6502 Development Pack £20, Hi-View £20. Also 5 Micropower games £15. Tel. (0256) 464889.

HELP WANTED. BBC B+ owner wants help with tape to disc, none so far successful. Any help appreciated. Second disc drive wanted, 40/80, must be reasonable. C. Butcher, 47 Clumber Drive, Gomersal, Cleckheaton BD19 4RP.

BBC B O.S. 1.2 & Wordwise Plus ROM handbook £30, Elite Disc £10, View Calc tape £10, View File tape £10, Kensington tape £10, Painting tape £5, Drawing tape £5, Greedy Dwarf tape £5, Complete set of BEEBUG Vol's 1 to date £30, 30 Hour Basic £5, Advanced User Guide £10. Tel. (0272) 784250 (eves. only).

BBC 'B' with DNFS, ATPL ROM Board, Acorn Speech, Vine Micros Replay Tape/Disc ROM, £150. Acorn Z80 2nd processor with all s/ware and manuals £150. Dual D/S 40/80 disc drives (MITZU) £150. Microvitec med. res. colour monitor £175. WW+ £35, Toolkit Plus £18, Disc Doctor £18, AMX Super Art & Mouse £50, AMX Pagemaker £25, Quinkey & s/w £25. Tel. Stevenage (0438) 811823 (eves.).

BBC Master 128, 400/640K disc drive, Interword, ADT, MAX, AMX Super Art & Mouse, Cartridge, Joystick, Office Mate and Office Master, 50+ games Elite, Repton 3, Sentinel etc, and all manuals. Cost £1000. bargain only £650. Tel. (0642) 786468.

BBC B TAPES: White Knight, Frogger, Morse, Drawing, Creative Graphics, Chemiplant, other educational tapes, £1 to £4 each. Tel. 01-500 7476 after 5pm.

Viglen Console for Master 128 complete with power supply unit. Bargain at £40. Sorry, no offers. Tel. Cooden (04243) 4500.

System Delta with Inter/View Link & Advanced Reference Manual £45, Disc software: Stryker's Run £6, Grand Prix Construction Set £6, Graphic Adventure £12, The Hobbit £7, BBC Bonanza compilation £9, Dr Who & the Mines of Terror £8. Tel. (0242) 513264.

BBC software on tape games and serious software all at reasonable prices. Can be bought as a whole or separate. Tel. Hull (0482) 640654 (eves.).

Microvitec 1451 Monitor, nearly new, £190. 6502 second processor, as new, £90. ATPL Sidewise board with RAM £30. Unopened BBC Monitor ROM £15. Tel. 01-524 2539.

Wordwise Plus ROM with Bruce Smith's book £25, Watford ROMSPELL with manuals £18, or both for £39. Tel. (0234) 750050.

Master Compact with second drive, Microvitec 1451 colour monitor, RS232, joystick, Mertec Companion upgrade, Viglen ROM cartridge adaptor, printer cable. As new £700. Tel. Oxford (0865) 739766.

BBC ROMs, MAX £10, Disc Doctor £12, Computer Concept's Graphics ROM £10, Castle Quest, Fortress, Frenzy on disc £3 each. Casio FX-702P calculator with printer and cassette interface £65. Tel. 01-960 8820.

WANTED Torch Unicorn for BBC. Tel. 01-206 0544.

Dataproducts professional daisy wheel printer model B50 with handbook plus lots of accessories, e.g. single-sheet feeder, ribbon, daisy wheels, etc. £375 ono. tel. (0727) 66849.

TYPEASY typing course for BBC, new £6. Epson MX80 dot matrix printer in VGC, £90. Viglen perspex printer stand 80 col, new £12. Tel. (0234) 750050.

Aries B32 and B12 with 16K RAM, boxed with instruction manuals, £85. Tel. Chesterfield (0246) 74436.

BBC Master 128 brand new only £335. Tel. Bedford (0234) 67067 (eves.).

SPECIAL OFFER! — £5.00
 of ADE+ excluding Technical
 Reference Guide. Still available at
 £12.00 inc. VAT

At last a machine code development system that *really* does outperform ADE

SYSTEM Software

Dept. G
 Fifth Floor Sheaf House
 Sheaf Street
 Sheffield S1 2BP
 Tel. (0742) 768632

Access and Barclaycard welcome
 Further information available



Features

- Runs on BBC B, B+, Master, Compact, ADFS, DFS, ANFS, NFS
- Full use of all available RAM
- Intelligent memory management unit (MMU)
- User switches on all main features
- Automatic search for macros on disc
- Assemble absolute or relocatable code
- Source program tokenised or full text
- REPEAT, UNTIL, WHILE, WEND etc. high level language constructs
- Macros nestable to any depth
- Excellent error diagnostics
- Linker
- Editor and symbolic disassembler provided

A comparison of three 65C12 assemblers

Feature	ADE +	BBC Basic	MACROM
Number of pseudo-ops	64	4	36
Use of all available RAM	Yes	No	No
Macros	Yes	No	No
String functions	Yes	Yes	No
True macro libraries	Yes	No	No
Relocatable output	Yes	No	No
Linker with libraries	Yes	No	No
High level constructs	Yes	No	No
Use any editor?	Yes	No	No
Number of error reports	39+warnings	Yes	20
65C00 extended opcodes	All	65C12 only	65C12 only
Switch off 65C12 opcodes	Yes	No	No
ROM size	32K	16K	16K
Disassembler	Symbolic	No	Simple
Label restrictions	No	Yes	Yes



Versions and prices

Recommended for Master Compact.
 ADE+, MMU and 65C00 series assembler on disc
 (3.5" ADFS). 32K sideways RAM required.
£42.00 + vat

Recommended for BBC B, B+.
 ADE+, MMU and 65C00 series assembler on 2
 16K EPROMs with DFS 5.25" utilities disc.
£46.00 + vat

Recommended for Master 128, turbo.
 ADE+, MMU and 65C00 series assembler on
 EPROM cartridge with 5.25" DFS utility disc.
£49.00 + vat

Upgrade
 ADE+ to ADE+ (upgrade to either disc,
 EPROM or cartridge, you must send in ADE
 ROM chip)
£27.00 + vat

Upgrade
 ASM to ADE+ (upgrade to either disc,
 EPROM or cartridge, you must send in ASM
 ROM chip)
£35.00 + vat
 Please add £1.25 P&P per unit.



The Ultimate Assembly Language Development Tool

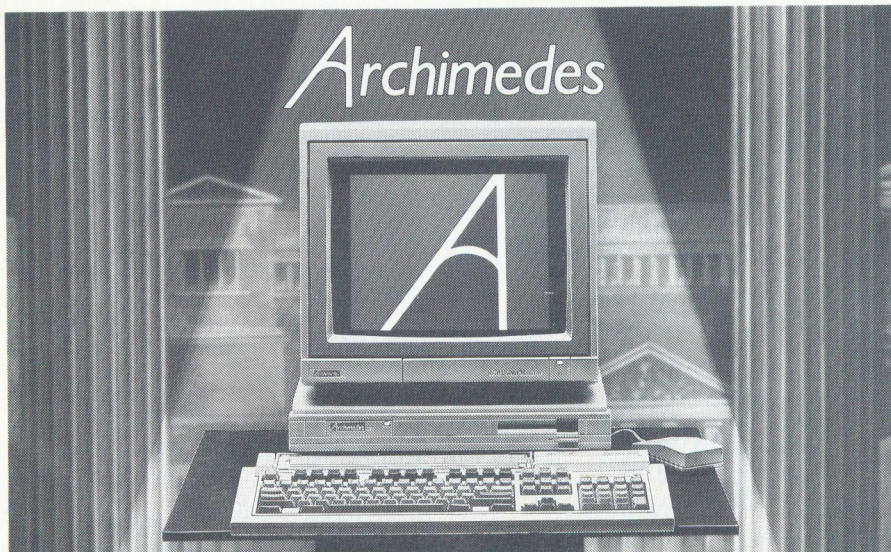
ADE+ is a 65C00 series assembler system supporting all the mnemonics of the 65C12 used in the latest BBC microcomputers plus the additional 'Rockwell' instructions. ADE+ is fast, faster in fact than the in-built BASIC assembler and all rival products that we have tested. The assembler produces absolute code that can be

the output from other programs using the ADE+ linker. ADE+ supports a powerful linker which drastically cuts assembly time; a feature normally only found on minis and mainframes. The linker will even link the output from compilers with your assembly language programs. Full library support for both the linker and the assembler is provided - fast searching for unknown instructions in a random access macro library. ADE+ is a modular system with many modules to add later, i.e. a mouse based editor & a Z80 cross assembler! A print spooling system uses sideways RAM as a print buffer to eliminate waiting time; your listing runs off as a background job! Use the print spooler from BASIC or your own programs. ADE+ uses ALL available memory. With a second processor attached the IO processor spare memory is used as a buffer to reduce the amount of disc access. All available memory is handled by ADE+'s intelligent memory management module. Use your own favourite editor or the one provided. Assemble from disc or memory. Full utilities including librarians, converter for BBC BASIC etc. ADE+ must be the bargain of 1986/7!

It operates at
4 million instructions
per second.

It's the world's fastest
microcomputer.

It has been voted
the 1987
microcomputer
of the year.[†]



†Home/Small Business Category of the British Microcomputing Awards.

Model shown here is the Archimedes 310 with colour monitor and includes Mouse.

- ▷ At an operating speed of 4 million instructions per second just about everything you do happens instantly. With this kind of power at your fingertips the advantages are enormous.
- ▷ Already Archimedes* has won the British Microcomputer of the Year award, a clear winner against the other two finalists, the Amstrad 1512* and the Amiga 500*
- ▷ In addition to BBC Basic V, high level languages such as 'C' and 'Pascal' can be used for specialist applications fully exploiting the computer's outstanding power, graphics and speed.
- ▷ The huge potential of Archimedes doesn't stop there. The 310M with its built-in PC emulator can run Lotus 1.2.3*, D BASE III*,

MS-WORD*, and other popular business programs available under MS-DOS*

- ▷ With prices starting at £749 exc. VAT for the 305 with mono monitor and rising to £1035 exc. VAT for the 310M with colour monitor, the Archimedes 300 series represents unparalleled value for a computer system of such capability.

PRIORITY REQUEST

Please send me an information pack on Archimedes together with a list of dealers.

Name: _____

Address: _____

Postcode: _____ Tel: _____

☐ Tick here for written details of the 0% finance deal available through participating dealers.

Acorn Computers Ltd.,
FREEPOST 4335, BS1 3YN

Acorn
The choice of experience.
16

— OR PHONE 0800 100 100 —

0% (APR 0%) FINANCE FOR LIMITED PERIOD

Interest free credit is available on all Archimedes systems during the period 1.10.87 to 31.12.87. Maximum period of credit 12 months. Subject to status. Written quotation available on request from participating dealers who are licenced credit brokers.

*MS-DOS and MS-WORD are trade marks of Microsoft Corporation.

*Amiga 500 is a trade mark of Commodore Business Machines.

*Amstrad 1512 is a trade mark of Amstrad Consumer Electronics plc.

*Lotus 1.2.3 is a trade mark of Lotus Corporation.

*D BASE III is a trade mark of Ashton-Tate.

*Archimedes is a trade mark of Acorn Computers Ltd.

UK BULLETIN BOARDS

Continued from Vol.6 No.6

RSGB 0707 57477
24 Hours (Hertfordshire)
1200/75 Viewdata

SABBS 0698 884804
24 Hours (Larkhall)
300/300

SBBS The Irishman 0247 455162
21.00-09.00 (N.Ireland)
300/300

SBBS Watford 0923 676644
21.00-08.00 (Hertfordshire)
300/300

SBBS Stevenage 0438 722128
24 Hours (0800-
2300 ringback) (Stevenage)
300/300

Stoke ITEC 0782 265078
24 Hours (Stoke-on-Trent)
1200/75 Viewdata

Swafax 0622 850440
24 Hours (Kent)
1200/75 Viewdata

TBBS Blandford 0258 54494
24 Hours (Dorset)
300/300

TBBS London 01 348 9400
24 Hours (London)
300/300 & 1200/75

TBBS London NW 01 200 7577
24 Hours (London)
300/300 & 1200/75

TBBS Rovereed 01 542 4977
24 Hours (London)
300/300

TBBS West Midlands 0384 635336
18.00-08.30 (London)
300/300

Techno Fresh Systems 0570 423082
24 Hours (Dyfed)
300/300 1200/75

Technoline 01 450 9764
24 Hours (London)
1200/75 Viewdata

Telemac 15, OBBS 0625 33703
24 Hours 300/300

Timestep 0440 820002
(Suffolk) 1200/75

TUG London 01 200 7577
24 Hours (London)
300/300

TUG Board II 021 444 1484
20.00-08.00 (Birmingham)
300/300

WBBS Wimbledon 01 542 3772
WE 19.00-02.00 (London)
1200/75 & 300/300

Additional Boards

Club 1512 01 204 8755
24 Hours 1200/75 300/300

EBBS No.1 0274 541156
24 Hours (Bradford)
1200/75 300/300

Enterprise BBS 0482 868388
after 10pm or
ring & request Beverley, Humberside
1200/75

HBBS Aberdeen 0224 632570
24 Hours (Aberdeen)
1200/75 300/300

continue on page 40

Please notify us of any changes or omissions to the information published here.

UPDATE

USER GROUPS

BOCHUM West Germany GERACUS

German Acorn User Club.

Contact:

Roul Sebastian John,
Wassertrasse 475,
4630 Bochum 1,
West Germany

BIRMINGHAM

A new user group is
being formed as from
12th January next. It will
meet on the second
Tuesday of each month at
the Headquarters of the
Midland Radio Society,
Unit 5 Henstead House,
Henstead Street,
Birmingham 5.
Contact Michael Nyman
on 021-382 3606 for more
information.

SPECIAL OFFER POLAROID PALETTE FOR THE B.B.C. MICRO

▼ **£490** (+ VAT)

System includes
Autoprocessor,
Cutter Moulder,
Sample Films.

*The simple way to
instant 35mm
presentation
slides, prints
and overhead
transparencies.*



To take advantage of this limited period offer please send
your order to: Robin Cartwright Polaroid (U.K.) Ltd.,
Ashley Road, St. Albans, Hertfordshire AL1 5PR.

- ☐ Please debit my Polaroid Account No:.....
☐ Cheque for full order value enclosed (£563.50 inc VAT).....
☐ Please debit my Access Account No:.....
 Barclaycard Account No:.....

Name..... Address.....

Signature.....

ELECTRIC PROCESSOR/ILLUMINATED CUTTER MOUNTER £50 EXTRA (+ VAT)

Polaroid

Polaroid is a registered trademark of Polaroid Corporation, Cambridge, Mass. USA

ADVERTISING IN BEEBUG

For advertising details, please contact
Yolanda Turuelo
on
(0727) 40303
or write to
Dolphin Place, Holywell Hill,
St Albans, Herts. AL1 1EX

Fast Flexible Professional Software

Excellent software at unbeatable prices.
AVAILABLE NOW.

A R C H I M E D E S

System DeltaPlus £69.95 WIMP Database Management System, 2 billion records, multi-tasking with multiple files open at once. Multiple subfiles. Instant card find. Comprehensive maths searches and sorts. Multiple user-definable print formats. Mailing labels. Card copier. Imports files from dBase. Viewstore, etc. Programming language included.

DataBase £29.95 Affordable excellent. Searches, sorts, maths. Alternative print formats. Import data from other databases.

System Delta Business Accounts £69.95 each module The only comprehensive, flexible accounting suite. Use records, pull down help screens. Searches, sorts, maths. Alternative print formats. Import data from other databases.

System Delta Business Accounts £69.95 each module The only comprehensive, flexible accounting suite. Use records, pull down help screens. Searches, sorts, maths. Alternative print formats. Import data from other databases.

System Delta Business Accounts £69.95 each module The only comprehensive, flexible accounting suite. Use records, pull down help screens. Searches, sorts, maths. Alternative print formats. Import data from other databases.

COMING SOON:
System Gamma
System Sigma

B B C M I C R O

System Gamma with Card Index £64.95 Still the most comprehensive and flexible data management system on the BBC. Friendly user Card Index. 8160 records, 255 fields. Finds unique record in 3 secs. Versatile search, maths & sort by any criteria. Add fields/change layout after entry of records. Programming charts easily from the menu driven GammaPlot supplied. Multiple charts and text on screen at any one time and in any position desired. Various text fonts. Statistical analysis. Macro facility. Entry by keyboard or System Delta files. Programming language included.

System Delta Business Accounts Order Processing/Invoicing Sales Ledger, Stock Management, Purchase Ledger, £45.95 each, Nominal £55.95

Database Utility Software £19.95 each

Specialist Applications School Administrator, Hotelier, Video Rental and Newsagent, from £54.95

Access & Barclaycard accepted

MINERVA SYSTEMS

69 SIDWELL STREET · EXETER · DEVON EX4 6PH · TELEPHONE EXETER 0392 37756

International Distributors in Africa, Australia, Belgium, Canada, Cyprus, Greece, Holland, Hong Kong, Iceland, New Zealand, Malta, Norway, Portugal, Sweden, U.A.E.

Business Ads

THE SCRAMBLER FOR BBC B, Master and Compact. The system that will help you comply with the Data Protection Act. ENCODE your disc files of any size or type from within SCRAMBLER's comprehensive DFS/ADFS Menu system. Your own unique PASSWORD (supplied under separate cover). Ideal for schools. £19.95 (p&p inc). Supplied on ROM. *Number 3 Software, 3 Dairy Farm Court, Attleborough, Norfolk NR17 2BT.*

continued from page 38

HBBS No.1
24 Hours
0274 42957
(Bradford)
1200/75 300/300

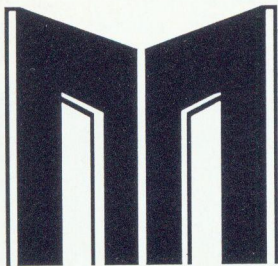
Hull BBS PC System
24 Hours
0482 20909
(Hull)
1200/75 300/300

Infocom
24 Hours
021 476 9881
(Birmingham)

Intaview
24 Hours
1200/75 300/300
021 622 5010
(Birmingham)
1200/75

Madhouse
24 Hours
061 477 8405
(Manchester)
1200/75 300/300

Please notify us of any changes or omissions to the information published here.

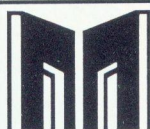


THE MASTER PAGES

Devoted to the Master Series Computers

We have another useful utility for Master and Compact users this month in the form of a fast and efficient back-up facility for those with a single disc drive. A further selection of hints and tips rounds off our Master pages. Note, however, the Modular Assembler which follows the Master pages, as this can exploit the additional memory of the Master and Compact to good effect.

Hints & tips, programs and articles, short or long, will be most welcome for publication in our Master pages. Let's hear from you.



**MASTER
SERIES**
**SINGLE DRIVE
ADFS BACKUP**

P.J.Short presents a useful utility that enables an ADFS disc to be backed up quickly and efficiently on a Master or Compact using only a single drive.

Making backup copies of double-sided 80 track ADFS format discs is a long and tedious activity. It involves over 80 disc swaps, as the standard backup utility uses a relatively small buffer area to hold the data being copied. This program is designed to

speed up and simplify the task on a BBC Master or Compact, by using a large proportion of all available RAM - main, sideways and shadow - as a buffer. By copying the disc content in "chunks" of 96 Kbytes, it is possible to back up a double sided 80 track disc with only 15 disc swaps.

USING THE PROGRAM

When the program is run you are asked to place the source disc in drive 0 and then press the space bar. The program reads the size of the disc (40, 80 or 160 tracks) and the title of the root directory, both of which it displays in a screen window. If the disc appears to be empty (a zero value for the master sequence number on the disc), you will be prompted to confirm that the backup should continue.

You now place the target disc in the drive and press the space bar again. The program checks the formatted size of this disc and flags any difference from that of the source disc. The target disc must be the same size and format as the source disc, and should be replaced if a difference is reported. If the target disc appears not to be empty (a non-zero value for the master sequence number), you will be asked to confirm the intention to overwrite its existing contents.

The process of copying from the source to target disc now commences with disc changes being prompted as required. The progress of the back-up process is shown by a display of the numbers of the tracks currently being read from or written to. When copying is completed you will be offered the choice of backing up another disc or leaving the program.

PROGRAM NOTES

The drive on which copying takes place is preset to 0 in line 150. The program uses the ADFS general read/write routine (OSWORD &72) to access blocks of 16k bytes on disc. The routine is called via the Basic function FNsector, to read from the source disc into a buffer area in either main or shadow memory. Data is written from the buffer area to the target disc. The reads and writes alternate as back-up proceeds.

Single Drive ADFS Disc Backup

Source disc is 160 track ADFS format.

Root title is Dave's Work Disc

Place target disc in drive 0

Press <SPACE> when ready

As noted earlier, the program copies "chunks" of up to 96 Kbytes at a time from source to target disc. The sequence for achieving this is as follows:

1. Read a 16k byte block into a buffer in main memory.
2. Copy the contents of the buffer into sideways RAM bank 4 using *SRWRITE.
3. Repeat operations (1) & (2) three times, but using RAM banks 5, 6 & 7 successively.
4. Read the next block into main memory.
5. Read a further block into shadow memory, selected by *FX108,1.
6. This block is then the first to be written to the target disc, at the correct location relative to its position in the 96k byte "chunk".
7. Main memory is re-selected by *FX108,0 and the block held there written to the target disc.
8. The content of sideways RAM bank 7 is copied back to the main memory buffer using *SRREAD and from there to the target disc.
9. Cycle 8 is repeated successively for RAM banks 6, 5 & 4.

Note that HIMEM is set to &3000 to provide room for the buffer areas in main and shadow memory. In principle HIMEM could be set to &3C00 and still provide a 16 Kbyte buffer. However, this could lead to uncertainty regarding the location of the Basic stack. With HIMEM at &3000, the Basic stack is guaranteed to be in main memory.

```

10 REM Program ADFSback
20 REM Version B1.23
30 REM Author P.J.Short
40 REM BEEBUG December 1987
50 REM Program subject to copyright
60 :
100 MODE7
110 buffer=&3000
120 HIMEM=buffer
130 DIM osblock 15
140 osword=&FFF1
150 drive=0
160 PROCscreen
170 ON ERROR PROCerror
180 REPEAT:REPEAT
190 maxblk=FNnitsource(drive)
200 UNTIL maxblk>0
210 REPEAT UNTIL FNcheckdest(drive,max
blk)
220 PROCbackup(drive,maxblk)
230 PROCwindow1
240 CLS
250 PRINTTAB(0,1)"Job completed"
260 UNTIL NOT FNconfirm("Backup anothe
r disc?")
270 VDU26,12
280 END
290 :
1000 DEF PROCerror
1010 CLS
1020 IF ERR<>17 PRINTTAB(0,2)::REPORT
1030 PROCwindow1
1040 CLS
1050 PRINTTAB(0,1)"Job aborted/failed"
1060 IF FNconfirm("Exit program?") THEN
VDU26,12:END
1070 CLS
1080 PROCwindow2
1090 CLS
1100 PROCwindow3
1110 CLS
1120 ENDPROC
1130 :
1140 DEF FNconfirm(prompt$)
1150 LOCAL key
1160 PRINTTAB(0,4)prompt$CHR$134"<Y/N>"
CHR$135;
1170 *FX21
1180 REPEAT
1190 key=GET AND &DF
1200 UNTIL (key=89) OR (key=78)
1210 CLS
1220 =(key=89)
1230 :
1240 DEF FNnitsource(drive)

```



```

1250 LOCAL sectors,name$,sequence
1260 PROCmountsource(drive)
1270 IF FNsector(buffer,FALSE,drive,0,7
*256)<>0 PROCreadererror
1280 sectors=(buffer!252) AND &1FFFFF
1290 name$=""
1300 FOR I%=0 TO 17
1310 name$=name$+CHR$(buffer?(I%+512+12
41))
1320 NEXT
1330 sequence=buffer?512 OR buffer?(512
+1274)
1340 PROCwindow1
1350 CLS
1360 PRINTTAB(0,1)"Source disc is"CHR$1
31;(sectors DIV 16)CHR$135"track ADFS""
format."TAB(0,4)"Root title is"CHR$131na
me$;
1370 IF sequence=0 PROCwindow2:PRINTTAB
(0,1)"Source disc appears to be empty!":
IF NOT FNconfirm("Continue backup?") THE
N =0
1380 =(sectors-1) DIV 64
1390 :
1400 DEF FNcheckdest(drive,maxblk)
1410 LOCAL sectors,name$,sequence
1420 PROCmountdest(drive)
1430 IF FNsector(buffer,FALSE,drive,0,7
*256)<>0 PROCreadererror
1440 sectors=(buffer!252) AND &1FFFFF
1450 name$=""
1460 FOR I%=0 TO 17
1470 name$=name$+CHR$(buffer?(I%+512+12
41))
1480 NEXT
1490 sequence=buffer?512 OR buffer?(512
+1274)
1500 IF ((sectors-1) DIV 64)<>maxblk TH
EN PRINTTAB(0,1)"Target disc is"CHR$131;
(sectors DIV 16)CHR$135"track ADFS""for
mat, ie the wrong size.":PROCspace:=FALS
E
1510 IF sequence<>0 THEN PRINTTAB(0,1)"
Target disc is not empty!""Root title i
s"CHR$131name$:IF NOT FNconfirm("Overwri
te contents?") THEN =FALSE
1520 =TRUE
1530 :
1540 DEF PROCscreen
1550 ?&D0=?&D0 OR 2
1560 FOR I%=0 TO 24
1570 PRINTTAB(0,I%)CHR$132CHR$157CHR$13
5TAB(36,I%)CHR$132CHR$157CHR$32CHR$156;
1580 NEXT
1590 ?&D0=?&D0 AND &FD

```

```

1600 PRINTTAB(4,0)CHR$141"Single Drive
ADFS Disc Backup"TAB(4,1)CHR$141"Single
Drive ADFS Disc Backup";
1610 J%=3
1620 REPEAT
1630 FOR I%=J% TO J%+5
1640 PRINTTAB(3,I%)CHR$156
1650 NEXT
1660 J%=J%+7
1670 UNTIL J%>23
1680 ENDPROC
1690 :
1700 DEF PROCwindow1
1710 VDU28,4,8,35,3
1720 ENDPROC
1730 :
1740 DEF PROCwindow2
1750 VDU28,4,15,35,10
1760 ENDPROC
1770 :
1780 DEF PROCwindow3
1790 VDU28,4,22,35,17
1800 ENDPROC
1810 :
1820 DEF PROCspace
1830 PRINTTAB(0,4)"Press"CHR$134"<SPACE
>"CHR$135"when ready";
1840 *FX21
1850 REPEAT UNTIL GET=32
1860 CLS
1870 ENDPROC
1880 :
1890 DEF PROCmountsource(drive)
1900 PROCwindow2
1910 PRINTTAB(0,1)"Place source disc in
drive ";drive
1920 PROCspace
1930 ENDPROC
1940 :
1950 DEF PROCmountdest(drive)
1960 PROCwindow3
1970 PRINTTAB(0,1)"Place target disc in
drive ";drive
1980 PROCspace
1990 ENDPROC
2000 :
2010 DEF PROCbackup(drive,maxblk)
2020 LOCAL block,top
2030 block=0
2040 REPEAT
2050 PROCmountsource(drive)
2060 top=block+6
2070 IF top>maxblk THEN top=maxblk+1
2080 PRINTTAB(0,2)"Reading from tracks
";(block*4)" to ";(top*4-1);

```



```

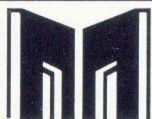
2090 PROCgetchunk(block,maxblk,drive)
2100 CLS
2110 PROCmountdest(drive)
2120 PRINTTAB(0,2)"Writing from tracks
";(block*4)" to ";(top*4-1);
2130 PROCputchunk(block,maxblk,drive)
2140 CLS
2150 block=block+6
2160 UNTIL block>maxblk
2170 ENDPROC
2180 :
2190 DEF PROCgetchunk(startblk,maxblk,d
rive)
2200 LOCAL block,ram
2210 block=startblk
2220 ram=4
2230 *FX108,0
2240 REPEAT
2250 PROCgetblock(buffer,drive,block*64
)
2260 IF ram<=7 PROCputtoSRAM(buffer,ram
)
2270 ram=ram+1
2280 IF ram>8 THEN *FX108,1
2290 block=block+1
2300 UNTIL (block>maxblk) OR (block>sta
rtblk+5)
2310 *FX108,0
2320 ENDPROC
2330 :
2340 DEF PROCputchunk(startblk,maxblk,d
rive)
2350 LOCAL block,ram
2360 block=startblk+5
2370 ram=9
2380 *FX108,1
2390 IF block>maxblk ram=ram-block+maxb
lk:block=maxblk
2400 REPEAT
2410 IF ram<9 THEN *FX108,0
2420 IF ram<=7 PROCgetfromSRAM(buffer,r
am)
2430 PROCputblock(buffer,drive,block*64
)
2440 ram=ram-1
2450 block=block-1
2460 UNTIL (block<startblk)
2470 ENDPROC
2480 :
2490 DEF PROCputtoSRAM(address,bank)
2500 OSCLI("SRWRITE "+STR$~address+" +4
000 8000 "+STR$bank)
2510 ENDPROC
2520 :
2530 DEF PROCgetfromSRAM(address,bank)

```

```

2540 OSCLI("SRREAD "+STR$~address+" +40
00 8000 "+STR$bank)
2550 ENDPROC
2560 :
2570 DEF PROCgetblock(address,drive,sec
tor)
2580 IF FNsector(address,FALSE,drive,se
ctor,&3000)<>0 PROCreaderror
2590 IF FNsector(address+&3000,FALSE,dr
ive,sector+48,&1000)<>0 PROCreaderror
2600 ENDPROC
2610 :
2620 DEF PROCputblock(address,drive,sec
tor)
2630 IF FNsector(address,TRUE,drive,sec
tor,&3000)<>0 PROCreaderror
2640 IF FNsector(address+&3000,TRUE,dri
ve,sector+48,&1000)<>0 PROCreaderror
2650 ENDPROC
2660 :
2670 DEF FNsector(address,write,drive,s
ector,bytes)
2680 LOCAL A%,X%,Y%
2690 ?osblock=0
2700 osblock!1=address
2710 osblock?8=sector MOD 256
2720 osblock?7=(sector DIV 256) MOD 256
2730 osblock?6=((sector DIV 256) DIV 25
6) OR ((drive*32) AND &E0)
2740 osblock?9=(bytes-1)DIV256+1
2750 osblock!11=bytes
2760 IF write THEN osblock?5=&0A ELSE o
sblock?5=&08
2770 A%=&72
2780 X%=osblock MOD 256
2790 Y%=osblock DIV 256
2800 CALL osword
2810 =?osblock
2820 :
2830 DEF PROCreaderror
2840 LOCAL A%,X%,Y%
2850 A%=&73
2860 X%=osblock MOD 256
2870 Y%=osblock DIV 256
2880 CALL osword
2890 error=osblock?3
2900 IF error THEN $&100=CHR$0+CHR$199+
"Disc error "+STR$(error AND &7F)+" at :
"+STR$(osblock?2 DIV 32)+" / "+STR$(!os
block AND &1FFFF)+CHR$0 ELSE $&100=CHR$
0+CHR$222+"Channel"+CHR$0
2910 CALL &100
2920 ENDPROC

```

Hints



Hints



Hints

EDITORIAL RESCUE

Dennis Weaver

If your machine hangs when you are using the Editor, you can usually rescue your text by clearing the machine with Ctrl-Break, and then saving the contents of RAM with:

```
*SAVE filename E00 +n
```

You must replace n by the length of the file in hex. This only needs to be approximate, and can be wildly long if you are unsure of the length. Once you have made the save, turn off the machine to clear any nasties, and then execute

```
*EDIT filename
```

and use the Editor to clean up your file. You will need to check it through quite carefully, because it may well contain duplicated lines, which will need to be deleted.

DIRECT SCREEN ACCESS

Jonathan Wilkinson

When shadow memory is enabled, using *LOAD and *SAVE in the usual way to load and save screens will refer to main memory rather than screen memory. Using the ADFS, you can modify these commands as follows:

```
*LOAD <fsp> FFFE<nnnn>
```

where <nnnn> is the hexadecimal value HIMEM would assume if shadow memory was NOT enabled. The screen will then be loaded in correctly. Similarly:

```
*SAVE <fsp> FFFE<nnnn> +5000
```

will save screen memory to disc for modes 0, 1, and 2. Change the 5000 to suit other modes. These modifications will work whether or not shadow memory is enabled, and in any mode.

EDITORIAL HELP

Dave Somers

Also on the subject of the Master's editor, did you know it's possible to have on-screen help as an aide memoire? When in the editor, you just press Shift-f5 to change screen mode, except that if you enter D, the screen will change to mode 0 and the upper part will contain help text and a function keystrip. When you press any of the function keys, the upper display will then show details of the action that the function

key performs. If instead of selecting mode D, you select mode K, the screen display will change to mode 0 and a function key strip will appear at the top. Like some of the other options in the editor, these are stored in the CMOS RAM ready for when you next use EDIT.

MASTER ROM - ADT CLASH

David Graham

If you are using the Master ROM and also have ADT fitted in your machine, the *FCOPY command will not work properly if ADT is in a higher priority socket. You can get around this by making use of the Beebug "B" convention. Just preface commands directed to the Master ROM with a "B". Thus FCOPY becomes BFCOPY. Alternatively, you could reverse the priority of the two ROMs, but this will prevent access to ADT's FCOPY command.

MUSICAL TIMING

Jonathan Wilkinson

When using the M500/5000 music synthesiser with a Master, which also contains the ADT ROM, there can sometimes be problems. If the music goes out of tempo, then you should use *UNPLUG n where n is the number of the ROM socket containing the ADT ROM. (To find out which, execute *ROMS). Then press Ctrl-Break.

Note that these comments about ADT should not give you the idea that this ROM is poorly written. It is a well written, stable and most useful product.

APPENDING BASIC

Dennis Weaver

This short EXEC file can be used to append a Basic program to the one currently in memory. Ideally the EXEC file should be placed in the library directory of your disc, where it can be called by typing *APPEND (assuming you saved it as file "APPEND"), and that your library is assigned.

```
*KEYOVDU21|MVDU6:INPUT"Prog Name?"FI
LE$:A%=TOP-2:OS. ("LOAD "+FILE$+" "+STR$
~A%)|M
```

```
*FX15,1
```

```
*FX138,0,128
```


THE BEEBUG MODULAR ASSEMBLER

Barry Christie describes a modular assembler program which allows assembler source files to be split into manageable units. A special facility permits assembly into sideways RAM on a Master or Compact.

One of the problems with the Beeb's built-in assembler is that it is cumbersome to use when dealing with long source code programs. Firstly, you can easily run out of memory when assembling, because both source and object code must be held in memory at the same time. Secondly, very long source files can be quite cumbersome to use simply because of their length. The BEEBUG Modular Assembler gets around both of these problems by allowing you to split up your source code into small and manageable modules.

On a Master and Compact, the memory problem is further improved by making use of sideways RAM to hold the object code generated by the assembler. An added bonus of modular assembly is that it encourages a structured approach to assembly programming, and makes it easier to create libraries of commonly used code, which can be incorporated as separate disc-based modules in any number of different programs.

USING THE MODULAR ASSEMBLER

Since it is easiest to explain the way in which the program works by direct example, we will begin by getting the assembler up and running. First of all, type in the program and save it away. Next type in and save the two example modules listed as *Modul01* and *Modul02*. Each should be saved away under the name given. Modules have a common root name which is set up in the program, as described later. Now when you run the Modular Assembler, it will display a set of parameters, and wait for the space bar to be pressed.

When you do this it will begin to load in and assemble the two modules, giving assembly address and length of code for each module. If the assembly terminates without error, you will be prompted to press function key f8 to save the assembled code, and then, if you wish, to press f9 to load and run the

assembled code. If all goes well, the example modules should cause the following message to appear on the screen:

```
This text was assembled on module 1
This text was assembled on module 2
```

The assembler contains a reasonably sophisticated error trapping routine which should keep you apprised of any errors. Where appropriate, the full line in error will be listed for subsequent correction. Moreover, if the error occurs in one of the modules (rather than in the Modular Assembler itself), it will be individually loaded from disc, and again the error line will be listed out in full. To assist in debugging assembler modules, the first four function keys are set up as follows:

```
f0 Load in Modular assembler
f1 Save Modular assembler
f2 Load Module
f3 Save module
```

in the case of f2 and f3, you will need to supply the number of the module required (in two digit form - see later), together with an end quote (").

```
=====
USER RAM ASSEMBLER / Pass #
=====
Assembling      Address Len Code
=====
Source code filename: Modul
Object code filename: Object
Total source modules: 2

Code load address   : &3000
Execution address   : &3000

Press space to assemble
```

HOW IT WORKS

Essentially the program appends each module in turn to itself, assembles its contents, then deletes it and appends the next. Since it is necessary to perform a two-pass assembly, the whole process is repeated, with error signalling on the second pass only. As the code is generated, it is placed directly into RAM (above HIMEM on a model B, and at &C400 on a Master or Compact). On Master Series machines, the assembled code is then removed after each module has been assembled, and placed into sideways RAM for safe keeping. Two banks of SWR are used (see line 1150), one to temporarily hold all filing system data from the RAM area &C400, the other to hold the assembled code, before it is saved to disc at the end of the process.

TAILORING THE SYSTEM

If you are going to use the assembler for your own code, then you will need to insert your own data at two points in the program. The first is at lines 1030 to 1070. Here you should insert the root name of your source modules, the filename that you wish to be used for the object code, the number of active modules, and the load and execution address of the final code.

The other point at which data may need to be changed is at lines 1370 to 1410. This area holds any variables or addresses that any of your modules may need. As you can see, we have used 1380 and 1390 to define some of the most useful operating system entry points. I should stress that none of these are used by the Modular Assembler itself, though one, `oswrch`, is used in the example modules.

```
5000 \ MODULE 1
5010 \ Filename Modul01
5020 LDA #22
5030 JSR oswrch
5040 LDA #7
5050 JSR oswrch
5060 LDY #0
5070 .loop1
5080 LDA text1,Y
5090 BEQ out1
5100 JSR oswrch
5110 INY
5120 JMP loop1
5130 :
5140 .text1
5150 EQU $ "This text was assembled on M
odule 1"
5160 EQUW &0A0D
5170 EQUW 0
5180 .out1
5190 ]:ENDPROC
```

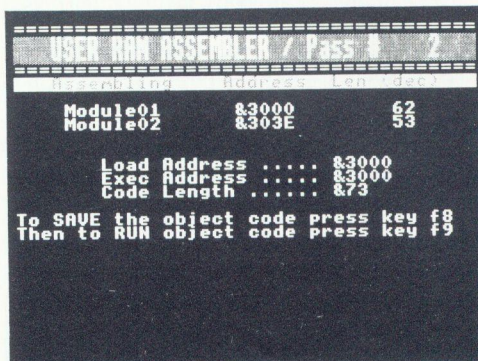
THE MODULES

Modules are very easy to create, as you can see from the examples. The rules are very simple. Each module must have line numbers which begin at 5000 or greater. Modules may use similar line numbers, or not, as you wish. Each module must terminate with the following:

```
] :ENDPROC
```

The filename of each module should consist of a root name (the same as that supplied by the user in the Modular Assembler itself), followed by two digits. These indicate the order in which the

modules are to be assembled, and must start with 01. The root name for the modules used in the example is *Modul*, and the two modules have filenames *Modul01* and *Modul02*. It is probably advisable to place the module name and number as a comment at the start of each module, as we have done in the example.



If you wish that a set of modules should form a continuous piece of code, as in the example, then be careful with the RTS instruction. There should be one of these at the end of your final routine, but you do not necessarily need one at the end of each module. All the code before the end-square bracket in each module should be written in assembler - even the very first line of a module. Thus all comments should be preceded by a "/" character, not a REM. Additionally, you must take care not to duplicate label names in different modules, since this will confuse the assembler.

```
5000 \ MODULE 2
5010 \ Filename Modul02
5020 LDY #0
5030 .loop
5040 LDA text,Y
5050 BEQ out
5060 JSR oswrch
5070 INY
5080 JMP loop
5090 :
5100 .text
5110 EQU $ "This text was assembled on M
odule 2"
5120 EQUW &0A0D
5130 EQUW 0
5140 .out
5150 RTS:] :ENDPROC
```


PROGRAM NOTES

The program automatically detects whether it is being used on a model B or a Master Series machine, and configures itself accordingly. The title banner indicates which mode of assembly is being used.

On the model B, as ever, memory may be a problem. If in doubt, keep your modules relatively short. It is possible to redistribute the amount of RAM used for modules and for the assembled code on a model B. Modules are stored up to the value of LOMEM, and the assembled code is stored above HIMEM. These two parameters are set up in line 110. On a model B, the object code can be a maximum of 4K in length, with the current settings. On a Master or Compact, the limit is 16K, with a maximum of 4K for the object code generated by each module.

Save this program under the name *ModAssC*.

```

10 REM Program Modular Assembler
20 REM Version B 1.0C
30 REM Author Barry Christie
40 REM BEEBUG December 1987
50 REM Program subject to copyright
60 :
100 Z%=INKEY(-256):IF Z%=253 OR Z%=245
THEN Z%=TRUE ELSE Z%=FALSE
110 MODE135:IF Z% THEN LOMEM=&5C00 ELS
E LOMEM=&4C00:HIMEM=&5C00
120 ON ERROR PROCerrormessage:VDU6:PRI
NT'':VDU26:PRINTTAB(0,20):END
130 SWR=Z%
140 PROCinitialise
150 PROCdisplay
160 PROCassembleprogram
170 PROCsaveobjectcode
180 PRINT'':VDU26:PRINTTAB(0,22);
190 END
200 :
1000 DEF PROCinitialise
1010 REM Replace the 5 values below wit
h your own data
1020 :
1030 source$="Modul"
1040 object$="Object"
1050 sourcemodules%=2
1060 realloadaddress$="&3000"
1070 executionaddress$="&3000"
1080 :
1090 *KEY0 LOAD"ModassC"|M
1100 *KEY1 SAVE"ModassC"|M
1110 OSCLI(" *KEY2 LOAD"+CHR$34+source$)
1120 OSCLI(" *KEY3 SAVE"+CHR$34+source$)

```

```

1130 OSCLI(" *KEY9 *RUN "+object$+"|M")
1140 IF SWR THEN SWR$="SIDEWAYS" ELSE S
WR$="USER RAM"
1150 IF SWR THEN filesocket$="6":codeso
cket$="7"
1160 procloadaddress%=TOP-2
1170 IF SWR THEN PROCfileinformation("S
RWRITE")
1180 ENDPROC
1190 :
1200 DEF PROCdisplay
1210 heading1$=" "+STRING$(38,"=")+" "
1220 heading2$=CHR$129+CHR$157+CHR$141+
CHR$135
1230 heading2$=heading2$+SWR$+" ASSEMBL
ER / Pass #"+STRING$(8,CHR$32)+CHR$156
1240 PRINTheading1$;heading2$;heading2$
;heading1$;
1250 PRINTCHR$135CHR$157CHR$129;" Assem
bling";SPC(4)"Address"SPC(2)"Len (dec)"S
PC(3)CHR$156
1260 VDU28,1,23,38,6:CLS
1270 PRINTTAB(1,2)"Source code filename
:";source$
1280 PRINTTAB(1,4)"Object code filename
:";object$
1290 PRINTTAB(1,6)"Total source modules
:";sourcemodules%
1300 PRINTTAB(1,10)"Code load address
:";realloadaddress$
1310 PRINTTAB(1,12)"Execution address
:";executionaddress$
1320 PRINTTAB(1,17)"Press space to asse
mble";REPEAT UNTIL GET=32
1330 PROCdefineaddresses
1340 ENDPROC
1350 :
1360 DEF PROCdefineaddresses
1370 REM User variables defined here
1380 oswrch=&FFEE:osbyte=&FFF4:osword=&
FFF1:oscli=&FFF7
1390 osfind=&FFCE:osbput=&FFD4
1400 :
1410 REM insert other variables here
1420 :
1430 realloadaddress%=EVAL(realloadaddr
ess$)
1440 IF SWR THEN sidewaysaddress%=&8000
-realloaddress%
1450 IF SWR THEN fakeloadaddress%=&C400
ELSE fakeloadaddress%=HIMEM
1460 ENDPROC
1470 :
1480 DEF PROCerrormessage
1490 ON ERROR OFF

```



```

1500 IF SWR THEN PROCfileinformation("S
RREAD")
1510 PROCchopoffmodule
1520 VDU 6:PRINT '
1530 IF ERR=17 THEN PRINT "<ESCAPE> ass
embly abandoned.":ENDPROC
1540 IF ERR=214 THEN PRINT "Module ";num
ber$;" not found.":ENDPROC
1550 REPORT:PRINT" at line ";ERL;" in "
;
1560 IF ERL<5000 THEN PRINT "program."
ELSE PRINT "module ";number$
1570 IF ERL>=5000 THEN OSCLI("*KEY4 LOA
D "+CHR$34+source$+number$+CHR$34+"|MLIS
T "+STR$ERL+"|M")
1580 IF ERL<5000 THEN OSCLI("*KEY4 LIST
"+STR$ERL+"|M")
1590 *FX138,0,132
1600 ENDPROC
1610 :
1620 DEF PROCchopoffmodule
1630 procloadaddress%?&00=&0D:procloada
dress%?&01=&FF
1640 ENDPROC
1650 :
1660 DEF PROCsaveobjectcode
1670 PROCchopoffmodule
1680 executionaddress%=EVAL(executionad
dress%)
1690 address1$=STR$~(realloadaddress%)
1700 address2$=STR$~(executionaddress%)
1710 IF SWR THEN address3$=STR$~(P%+sid
ewaysaddress%-&5000)
1720 PRINT"TAB(7)"Load Address ..... &
";address1$
1730 PRINTTAB(7)"Exec Address ..... &";
address2$
1740 PRINTTAB(7)"Code Length ..... &";
~P%-realloadaddress%
1750 PRINT"TO SAVE the object code pre
ss key f8"
1760 PRINT"Then to RUN object code pres
s key f9"
1770 IF SWR THEN PROCsidesave ELSE PROC
ramsave
1780 ENDPROC
1790 :
1800 DEF PROCassembleprogram
1810 FOR pass%=4 TO 6 STEP 2
1820 VDU28,35,3,35,1:PRINT ;pass% DIV 3
* 11;
1830 VDU28,1,23,38,6:CLS
1840 P%=realloadaddress%
1850 O%=fakeloadaddress%
1860 FOR modules%=1 TO sourcemodules%

```

```

1870 IF SWR THEN O%=fakeloadaddress%
1880 PROCloadassemblymodule
1890 PROCprocassemblmodule
1900 IF SWR THEN PROCsavesidewayscode E
LSE VDU6:PRINT P%-block%
1910 IF SWR THEN PROCfileinformation("S
RREAD")
1920 NEXT modules%
1930 NEXT pass%
1940 ENDPROC
1950 :
1960 DEF PROCloadassemblymodule
1970 number$=STR$(modules%):number$=STR
ING$(2-LEN(number$),CHR$48)+number$
1980 OSCLI("LOAD "+source$+number$+" "+
STR$~(procloadaddress%))
1990 ENDPROC
2000 :
2010 DEF PROCsavesidewayscode
2020 VDU 6:PRINT P%-block%:IF pass%=4 T
HEN ENDPROC
2030 sideways$="SRWRITE C400 "+STR$~(O%
)+" "+STR$~(block%+sidewaysaddress%)
2040 OSCLI(sideways$+" "+codesocket%)
2050 ENDPROC
2060 :
2070 DEF PROCfileinformation(loadsave%)
2080 OSCLI(loadsave$+" C400 D400 8000 "
+filesocket%)
2090 ENDPROC
2100 :
2110 DEF PROCsidesave
2120 $&A00="SRREAD 3000 "+address3$+" 8
000 "+codesocket$
2130 $&A40="SAVE "+object$+" 3000 "+add
ress3$+" "+address2$+" "+address1$
2140 *KEY8 VDU21|MX%=&00:Y%=&0A:CALL &F
FF7|MX%=&40:Y%=&0A:CALL &FFF7|MVDU6|M
2150 ENDPROC
2160 :
2170 DEF PROCramsave
2180 save$="SAVE "+object$+" "+STR$~fak
eloadaddress$+" "+STR$~O%
2190 $&A00=save$+" "+address2$+" "+addr
ess1$
2200 *KEY8 VDU21|M X%=0:Y%=&A:CALL &FFF
7|MVDU6|M
2210 ENDPROC
2220 :
2230 DEF PROCprocassemblmodule
2240 PRINTTAB(4)"Module";number$;SPC(6)
"&";
2250 PRINT STRING$(4-LEN(STR$~(P%)),"0"
);~P%;VDU 21:block%=P%
2260 [:OPT pass%

```


Dec McSweeney offers some suggestions for conserving the Beeb's limited memory.

People have praised the Beeb since it was launched way back when, but their paeans have almost always been qualified because of the lack of available memory. Incurable optimists see this as a blessing in disguise, presenting a challenge to their programming skills reminiscent of the steam-driven main-frames of their youth. The rest of us are going bald by degrees, tearing out another handful of hair at every dreaded "No room" error. This article is the first of two aimed at programmers conversant with Basic who write, or would like to write, large(r) programs. The techniques used in providing a solution to the problem use the so-called indirection operators ? and !. If you are unfamiliar with the use of these, I suggest that you refer you to an earlier Workshop Vol.3 Nos.7 & 8.

STORING NUMERIC VARIABLES

Numeric data is stored in one of two forms - integer (whole numbers) and real (numbers with a fractional part like 6.23). In BBC Basic, integers occupy 4 bytes, and real numbers 5 bytes. With integers, the 4 bytes allow a range between -2,147,483,648 to +2,147,483,647. If we were to reduce our range to -16,777,216 to +16,777,215 we would only require 3 bytes, -32768 to +32767 would require 2-bytes, and -128 to 127 would require 1 byte. Therefore, if your application uses a large table of small integers, for example screen co-ordinates, storing the data as a table of integers can be very wasteful of memory.

We can implement an array with any number of bytes per element using the following technique:

```
10 DIM table 200
```

This reserves 200 bytes for the array table, whereas the statement:

```
10 DIM table%(200)
```

reserves 804 bytes (201 integers requiring 4 bytes to store each one).

We access values from this new 200 byte array called table by using the byte indirection operator (?). This means that the expression:

```
X%=table(element)
```

is replaced by:

```
X%=table?element
```

(note that the first element is numbered 0).

The penalty for using this technique is that if we use the ? indirection operator, we will have to do some of the numeric manipulation ourselves. In BBC Basic, negative numbers are stored with a 1 in the most significant position. If we are to follow this technique with a single byte integer, the numbers 128 to 255 would represent the negative numbers -128 to -1. Fortunately, the storage of negative numbers is handled automatically by both the ? and ! operators. The same cannot be said of retrieval however, and the programmer must cater for this when using the ? operator. (The ! operator also handles the retrieval of negative numbers automatically.) A small piece of code will suffice for this:

```
195 REM Retrieving a (possibly) negative  
200 X%=?(table+element)  
210 IF X%>127 X%=X%-256
```

Numbers requiring more than one byte, but less than 4, require a slightly more complex approach. The Basic indirection operator ! can be used here, but it will convert a number into four-byte format, so care must be taken, when manipulating the table, to preserve the data in adjacent elements. A demonstration program will be found to accompany this article, and lines 1000 onwards are the routines for placing and retrieving two and three-byte values in the table. FNgetnum returns a specific element, while PROCputnum inserts X% into the table. Lines 100-270 demonstrate the functions in action.

As you will see, if you try changing the number of elements used (the variable max% in line 100), at least 12300 two-byte integers can be held in a

standard disc based BBC Micro, whereas a similar program, using a conventional table, gives "bad DIM" errors for over 6200 elements. A further advantage to this approach is that entire tables can be passed to procedures and functions as a parameter. This technique was covered in "Passing Arrays to Procedures" in Vol.5 No.2.

HOW THE ROUTINES WORK

Each element (n%) in the routines listed) occupies 2 (or 3) bytes, so the address of the element under consideration is 2*element or 3*element. The ! operator uses four bytes starting at each address, storing the least significant byte (LSB) first. The integer -1,820,300,768 (which is &93806A20 in hex) would be stored in succeeding bytes as &20,&6A,&80,&93. In the case of a two-byte table, this will overwrite the contents of the next element; a three-byte table affects the LSB of the next element. To preserve the value in the neighbouring element, we must construct a four-byte value which will be stored as B1 B2 B3 B4, where:

- B1 is the LSB of the number to be inserted
- B2 is the MSB (2-byte) or middle byte (3-byte) of our new number;
- B3 is the LSB of the next element (2-byte tables) or the MSB of a 3-byte tables.
- B4 is the MSB of the next 2-byte element or the LSB of the next 3-byte element.

This is achieved by ANDing our new number with &FFFF or &FFFFFF to zero the "top" part of the integer, then combining it with the byte(s) to be preserved using the OR function.

If you identify a need for both two and three-byte storage in the same program, you should rewrite the routines so that you can pass the element size to the procedure and function as a parameter. Really clever programmers like you and me could even store modest tables beneath PAGE or above HIMEM. I'll show you how next month.

Real numbers are more of a problem, but if your accuracy requirements are limited, say, to two decimal places for cash amounts, these should be converted to an integer by multiplying or dividing by the appropriate power of ten before storage, and a reciprocal factor on retrieval, eg:

```
225 REM converts pounds & pence to pence
230 PROCputnum(hourlyrate*100,element)
.
400 hourlyrate = FNgetnum(element)/100
405 REM reconverts
```

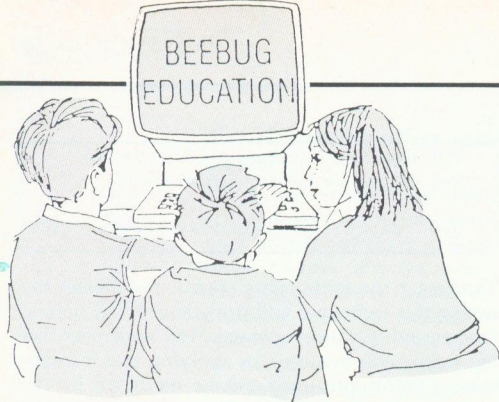
or:

```
300 degkelvin%=(tempcent+273.16)*1000
310 PROCputnum(degkelvin%,element)
.
900 degf=1.8*(FNgetnum(element)/1000)
```

The larger the table being stored, the greater the saving this technique will provide over the normal integer and real array storage. And remember that the extra code involved in applying this memory saving technique, may not be made up by the saving achieved in a small table.

```
60 REM Demo using 2-byte and 3-byte
70 REM numeric tables
80 REM By Dec McSweeney
90 :
100 max%=100
110 DIM table max%*3
120 CLS:PRINT"Put 2*N% in element N%"
130 FOR N%=0 TO max%
140 PROCputnum2(2*N%,N%):NEXT
150 PRINT"Reading & printing contents"
160 FOR N%=0 TO max%
170 PRINT ;FNgetnum2(N%); " ";:NEXT
180 PRINT"Press SPACE for 3-byte demo"
190 REPEAT UNTIL GET=32
200 PRINT"Put 100*N% in element N%"
210 FOR N%=0 TO max%
220 PROCputnum3(100*N%,N%):NEXT
230 PRINT"Reading & printing contents"
240 FOR N%=0 TO max%
250 PRINT ;FNgetnum3(N%); " ";:
260 NEXT:PRINT 270 END 280 :
1000 DEFPROCputnum2(X%,n%)
1010 table!(n%*2)=(X% AND &FFFF) OR
    (table!(n%*2) AND &FFFF0000)
1020 ENDPROC
1030 :
1040 DEFNgetnum2(n%):LOCAL a
1050 a=table!(n%*2) AND &FFFF
1060 IF a>32767 a=a-65536
1070 =a
1080 :
1090 DEFPROCputnum3(X%,n%)
1100 table!(n%*3)=(X% AND &FFFFFF) OR
    (table!(n%*3) AND &FF000000)
1110 ENDPROC
1120 :
1130 DEFNgetnum3(n%):LOCAL a
1140 a=table!(n%*3) AND &FFFFFF
1150 IF a>8388607 a=a-16777216
1160 =a
```

B



There's not a disc, tape or ROM in sight this month. Instead BEEBUG Education looks at the printed word. There are three book reviews, and a brief survey of the journal scene.

JOURNALS AND MAGAZINES

The standard magazine for several years in the field of computer education has been the monthly Educational Computing. It has recently changed format and publisher, and now comes out six times a year. This is twice a term and means that the news cannot be as recent as it can in, say, the Times Educational Supplement. The latter has also changed since BEEBUG Education was introduced, and there is now a weekly column as well as three substantial Computer Supplements (or Extras) a year.

Of an altogether more academic nature, is the now quarterly Journal of Computer Assisted Learning, issued by Blackwell Scientific Publications. There is inevitably an even longer time-lag here but the writers usually manage to provide interesting and often challenging insights into their specialities.

There is a group of publications from the specialist organisations such as MUSE, MAPE and BLUG. These seek to promote the interests of Secondary, Primary and Logo users respectively. They each publish journals, promote conferences and even sometimes software, and they clearly all have much to offer Acorn owners.

Lastly, if you have access to them, on-line databases such as Prestel (page 888) have lots of interesting information for educational users, particularly about tying products to uses, although they tend to be updated rather infrequently.

BOOKS ON EDUCATIONAL COMPUTING

Microcomputers and the Curriculum

edited by A.V. Kelly, published by Harper & Row at £6.50 (paperback), ISBN 0-06-318273-4.

This book is not as new as the other two reviewed here, but it is a major contribution to making the most of the micro in education, and has particular topicality into the bargain! Unlike some other purely descriptive titles around, it takes a definite line, and as such is very refreshing to read. Edited by Vic Kelly of Goldsmiths' College London, it surveys what is going on in schools and colleges, and the directions that we might take. It should really be read, as they say, by anyone new to the field, or anyone who wants to be aware of the major trends.

As in Kelly's other books, the curriculum and learning wisely come first. But the approach is not one of tacking computers on afterwards, but of exploring 'good practice' in the primary and secondary schools, and special sectors of education.

It largely takes the form of case studies. You may be weary of case studies which are little more than naive narratives. These can be of no real interest except to those involved. Only Beryl Maxwell's oft-printed "Why Logo?" chapter comes, perhaps, close to this category in places.

The chapters on overcoming computer-anxiety and on skills of hypothesising are particularly good. The techniques of "What would happen if.....?" are also dealt with by Deryn Watson in her chapter about the Humanities in Secondary Schools. This is an area where teachers convinced of the computer's value still find difficulty in introducing micros into their work. Read it, and then go off and get the software mentioned and run it on the Econet or BBC B stand-alones. David Dodds also stresses the need to work from existing good practice - this time in the Primary classroom - and gives more than a passing nod at the Plowden Report, its successors and at Cockcroft. This should mean, again, that teachers new or unsure about computers will feel at home.

In his own summary Kelly proposes that any overall influence which micros have in education may be in one of two forms: either as

a drill-and-practice teaching machine, or as a stimulus to imaginative teachers. In this preferable, second case the computer a much greater part in curriculum development than was envisaged, either by the original DTI scheme, or by the nonsense of a national or core curriculum.

Guide to Educational Computing
published by Surrey Publishing at £2.50
(paperback), ISBN 0-95-510360-76.

There is no such properly thought out theme in this publication. Rather, it is a collection of two dozen shorter essays, mostly relating specific products to their uses in schools. Indeed the book is full of advertisements. Is there any value in this, or is it just a commercial exercise? For anyone wishing to know what is happening in educational computing, this is one place to start.

How exhaustive is "Guide to Educational Computing" in the area, say, of Logo? Well, there is an article by Chris Roper of Logotron. Yes, they are market leaders. Yes, Logotron is the de facto standard in schools. But what is the title of the chapter? 'Educational Software' - misleading to say the least. The chapter on furniture (trolleys etc) is entirely devoted to that sold by one firm. In time, if you collected enough of these 'Guides' (yes, there are to be more), you would have an overview similar to one gleaned, for example, from scouring the advertisements in the BEEBUG supplement. As it is, I feel that although the information is useful and accurate, it is all too biased to be published under its present title. Be warned!

Trends in Computer Assisted Education
edited by Lewis & Tagg, published by Blackwell
Scientific Publications at £27.50 (hardback),
ISBN 0-632-01527-6.

There is no such shortage of substance in the last publication to be looked at this month. One of the briefs of BEEBUG Education was to look at all levels - not just Primary and Secondary. This book is intended as a potted survey of the use of computers Higher Education, being the Proceedings of the 1986 Computers in Higher Education Conference. Does this sound dry and academic? Don't be put off! The material gathered here is as varied and stimulating as in any compendium of its type.

The impression one is left with is of a lively community working at an ever broadening range of projects, and frequently making use of Acorn computers in the process. There are sections on Prolog and Logo, naturally, but also on interactive video, computer modelling including spreadsheets, with specific programs for the BBC B as well. For instance, there are program examples in BBC Basic illustrating the use of micros in investigative university mathematics.

I admit I was pleasantly surprised at how well Acorn micros are holding their own at this level. If proof was necessary that some of the most exciting and innovative developments don't need megabytes of memory, it is here in this book. There is, for example, a fascinating account and discussion of the design of a clinical simulation system at Liverpool University. Controlled interaction is the key. Like most of the material presented, here it is easy to read. The papers are short but in some specialist cases, background knowledge is a big advantage!

A good collection this, but the obvious drawback is the price. I guess it is inevitable with publications of this character. But if you are fascinated by the latest ways in which systems enhance learning at this level, do try to get a look at a copy somehow.

Useful information:

- BLUG** - British Logo User Group
PO Box 79, Walsall WS5 3RW.
- MUSE** - Microcomputer Users in Secondary Education
PO Box 43, Houghton on the Hill,
Leicester LE7 9GX.
- MAPE** - Micros and Primary Education
76 Sudbrooke Holme Drive, Sudbrooke,
Lincs.
- Harper & Row**, 28 Tavistock Street, London
WC2E 7PN.
- Blackwell Scientific Publications**, Osney
Mead, Oxford OX2 0EL.
- Surrey Publishing**, PO Box 56C, Esher, Surrey
KT10 9RA.

1st COURSE

Colouring your Beeb (part 2)

to do with the screen mode determined by the MODE command) can have any value in the range 0 to 4 as described in the User Guide (see table). Mode 0 is easy enough to understand, but what do the other modes mean, and what use are they anyway? In practice, mode 3 (Exclusive-OR) is the most useful plotting mode (after mode 0), but I am sure you will understand how the other plotting modes work as well by the time we have finished.

Mode	Meaning
0	Plot in specified colour
1	OR with existing colour
2	AND with existing colour
3	Exclusive-OR with existing colour
4	Invert existing colour

To illustrate matters, I want to introduce a program which we can add to as we go. It is given here in its simplest form (please keep to the line numbering given).

The purpose of this program is to 'draw' a red car and move it across the screen from left to right. The program uses mode 2, which allows a maximum of 16 logical colours to be used, useful as we shall see. The procedure PROCdefine creates two user-defined characters (using character codes 224 and 225) which when combined together form a passable likeness of a car (this is the variable car\$). User-defined characters are a subject in their own right, but this is reasonably well covered in the

Last month Mike Williams described how to use the GCOL command to select the foreground and background colours for your graphics. Now he takes a look at the different plotting modes, and shows how it is possible to add depth to your pictures.

The GCOL command requires two parameters:

GCOL mode, colour
as explained last month. The mode parameter (remember that this is the plot mode, nothing

important and useful technique in graphics programming, so we will examine the coding in some detail.

Program Car

```

100 MODE 2
110 PROCdefine
140 PROCmovecar
150 END
160 :
1000 DEF PROCdefine
1010 VDU23,224,0,0,63,51,115,127,127,24
1020 VDU23,225,0,0,224,48,63,255,255,12
1030 car$=CHR$224+CHR$225
1040 ENDPROC
1050 :
1060 DEF PROCmovecar
1070 VDU5:GCOL3,1
1080 FOR P=0 TO 1280 STEP 8
1090 MOVE P,300:PRINT car$;
1100 MOVE P,300:PRINT car$;
1110 NEXT P
1120 ENDPROC
    
```

The usual means of achieving the movement of an object on the screen is to display the object, then remove it, and move forward a short distance before repeating the process. This is a little like cine film which consists of a series of separate pictures. In both cases the changes must be sufficiently quick to fool the eye into seeing more or less continuous movement.

The procedure uses a FOR-NEXT loop to display the car in steps of 8 graphics units across the screen. To go with this, VDU5 causes the PRINT statement to 'print' at the current graphics cursor position (rather than the more usual text cursor), and allows us to use the MOVE statement to position the car finely. The thing that makes the whole process work, though, is the use of Exclusive-OR plotting (GCOL3,1 at line 1070). This mode has the effect that if an object is plotted twice in the same

position it first appears, and then disappears from the screen, exactly what we need. This is also the reason for the two identical statements at lines 1090 and 1100.

You will probably find that the moving car on your screen has a pronounced flicker. There are various ways of minimising this, but for now try adding:

```
1095 T=INKEY(5)
```

We will look in more detail at this problem in a future First Course.

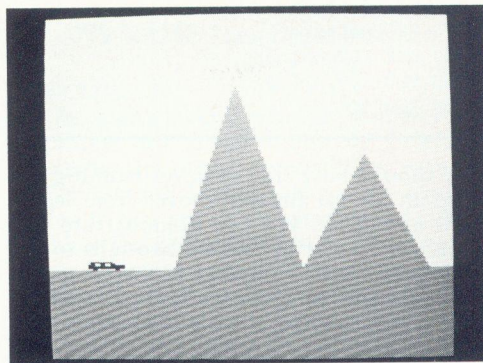
It is worth trying out this program at this stage to really appreciate the effect of what I have been describing. You may also like to experiment with changing the step size at line 1080, and discovering the effect of this for yourself.

At the moment you should see a red car moving across a black screen. Let's now add to the display by including some scenery. Add the following lines to your program, to define a procedure called PROCground and call this early in the program:

```
130 PROCground
1140 DEF PROCground
1160 GCOL0,2
1170 MOVE0,0:MOVE0,268
1180 PLOT85,1279,0:PLOT85,1279,268
1230 ENDPROC
```

If you run the program with these lines added, you should find your red car runs across a sward of green grass. You may wonder why the procedure PROCground uses a 'y' value of 268 when the car procedure uses 300. When characters are printed at the graphics cursor, the position referenced is the top left hand corner of a character. Each character is always eight pixels high, and one pixel is always equivalent to 4 graphics units vertically (the screen, remember, is 1024 graphics units from bottom to top). Thus the ground needs to be 32 graphics units below the position of the car (i.e. at a height of $300 - 32 = 268$). Understanding computer graphics often requires considerable attention to detail like this.

Let us now be more ambitious, and add some blue sky to our landscape. The following additional line will achieve this:



```
1150 GCOL0,134:CLG
```

This simply clears the whole graphics screen to background colour 134 (cyan blue) before any other plotting takes place. However, if you now re-run the program, you will find the car is no longer red, but white! Why is this? To answer the question we must begin to understand exactly what is meant by Exclusive-OR plotting.

Since there are 16 possible colours in mode 2, the logical colour numbers can be represented as four digit binary numbers. An Exclusive-OR operation on two binary numbers gives a binary result with a 'one' wherever either the first or second number (but NOT both) has a one, and a zero where both are one or both are zero. This is shown below for the two colours of red (colour 1 or 0001 in binary) and cyan (colour 6 or 0110 in binary):

red	0001	
cyan	0110	
Ex-OR	0111	white (colour 7)

The result of the operation is 0111 in binary, which is colour 7 (white). So how can we ensure that the car remains red even against the cyan? We could, of course, use GCOL0,1 when selecting red for drawing the car, but this would then destroy our technique for moving the car. The solution lies in the use of the VDU19 command that we used last month to change logical colour 7 to be red (colour 1). The car will then still appear in the required colour. To achieve this we will add one further procedure, PROCcolours, to our program to handle any colour changes of this kind that may be required. Here are the extra lines which will do this:


```

120 PROCcolours
1250 DEF PROCcolours
1260 VDU19,7,1,0,0,0
1300 ENDPROC

```

You should now find that on running the program the car (in red) moves from left to right as before. We can demonstrate this technique further by adding two hills to our landscape. These are added to the procedure PROCground and drawn with the PLOT command as follows:

```

1190 MOVE400,268:MOVE600,800
1200 PLOT85,800,268
1210 MOVE800,268
1220 MOVE1000,600:PLOT85,1200,268

```

If you re-run the program, you should now see a landscape with two green hills against a cyan blue sky, but we have the same problem when the car moves across the hills as previously with the sky. The colour of the car changes, and for the same reason - the Exclusive-OR operation on the two colours red (colour 1) and green (colour 2). The result of this operation is shown below:

red	0001	
green	0010	
Ex-OR	0011	yellow (colour 3)

Once again we can solve the problem by re-assigning colours using the VDU19 command, but now we have a choice - we can either make the car appear to move in front of the two hills (by changing colour 3 to be red) or we can make the car appear to disappear behind the two hills (by making colour 3 green). Take your choice, or even better try both. To make the car move in front of the hills add:

```
1270 VDU19,3,1,0,0,0
```

or to make the car run behind the hills add this variation of the same line:

```
1270 VDU19,3,2,0,0,0
```

But suppose we would like to be really clever and make the car run in front of one hill (say the first) and behind the other. We appear initially to be faced with a paradoxical situation as we cannot assign colour 3 (the result of the Exclusive-OR operation between red and green) to be both green and red. Yes, we could

perhaps try to capture the point when the car reaches a position between the two hills and change colour assignments at this instant, but this would be very tricky, and unnecessary.

What we have to do is to separate the two hills colourwise. The first hill will be drawn in colour 2 (which is green by default), and we shall now draw the second hill in colour 4 (chosen mainly because so far colour 4 is unused) to which we will assign the colour green with a VDU19 command. Both hills will then appear green even though they are drawn with different colour numbers. Modify line 1210 as shown (so that the second hill is drawn in colour 4), and add the second line (which changes colour 4 to be green):

```

1210 GCOLOR,4:MOVE800,268
1280 VDU19,4,2,0,0,0

```

If we want the car to move behind the first hill and in front of the second, then ensure that line 1270 reads:

```
1270 VDU19,3,2,0,0,0
```

and add one further line to change colour 5, the Exclusive-OR result of colours 1 and 4, to be red:

```
1290 VDU19,5,1,0,0,0
```

The result should be as we specified with the car moving across the screen, behind the first hill and in front of the second. Note too, that although only three colours appear on the screen, we have in fact used six of mode 2's sixteen logical colours. That's why mode 2 has been used in preference say to mode 5 which has only four logical colours.

Well, I did say at one point that computer graphics requires a deal of concentration. The way I have described the development of this program in fact follows very much the approach I took when creating the program in the first instance.

If at this stage you're still unsure about Exclusive-OR plotting, it may be worth re-reading this article, and in particular experimenting with the program and seeing if you can produce similar effects of your own. The complete program is also on this month's magazine disc/tape. More on colour next time in First Course.

ⓑ

CAMBRIDGE COMPUTER'S Z88 - A BEEB BY ANY NAME ? (continued from page 10)

IMPORT-EXPORT and TERMINAL

This option allows the transfer of files to and from the Z88. The manual describes the pin connections for the Z88 serial connector, and the protocol for file transfer. According to the manual, software is available for file transfer between the Z88 and the BBC micro, or IBM PC. However, with the details on connections and protocol provided in the manual, the software for handling the transfer at the BBC end can be easily written entirely in BBC Basic. The lead for the transfer can also be made by anyone competent at soldering. However, you should note that the manual is not too clear here, and you need to make a link between pins 8 and 9 in order to receive data.

To test the speed of transfer to a BBC micro I used the name and address file mentioned earlier containing 128 records. I found the maximum usable baud rate was 9600 (probably limited by Basic at the receiving end, so not the fault of the Z88). This resulted in a transfer time of 50 seconds, which considering the file contained some 19,000 characters is acceptable though it could hardly be called fast.

The Terminal application provides a widely used VT52 protocol for communicating between the Z88 and another computer, or between two Z88 computers, over the RS232 serial line.

SUGGESTED CONFIGURATION

I found that with the basic system (with no additional RAM), the memory is too limited, and I would therefore recommend that you skip the 32k RAM pack and purchase the 128k RAM. Also useful are either two sets of four rechargeable AA type batteries, or a mains power pack. The batteries last for about 20 hours actual use (over one year in a power off state). You can either make yourself a Z88 to BBC lead or buy the same. You will also have to decide whether you will be keeping backups of files by copying them via a BBC micro onto a floppy (the approach I take) or using an EPROM pack (you'll need two, plus an eraser for the maximum security), but the price is starting to soar. Prices of add-ons are given elsewhere in this review.

CONCLUSION

The Z88 goes a good step further than the Psion Organiser for those requiring briefcase portability. Its superb compatibility with the BBC micro and Master make it the obvious choice if you are in the market for such a machine. The Z88 needn't just be thought of for simply extending your Beeb, although it fulfils this role admirably. It should be thought of as a truly stand-alone system, capable of holding its own against the (albeit limited) opposition. I thoroughly recommend it.

PRICES (including VAT)

Z88 Computer	£287.50
Mains Adaptor	£9.95
32k RAM Pack	£19.95
128k RAM Pack	£49.95
32k EPROM Pack	£19.95
128k EPROM Pack	£49.95
EPROM Eraser	£39.95
Z88 to PC Link	£24.95
Z88 to BBC Link	£24.95
Serial Printer Cable	£9.95
Parallel Printer Cable	£29.95

B

Z88 SPECIAL OFFER

Because we feel that the Z88 is an excellent product, and has significant additional benefits for BBC Micro and Master users, we have put together a special package deal for BEEBUG members of:

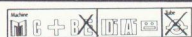
**Z88 computer,
Mains adaptor
128k RAM pack,
Z88/BBC transfer lead and
software.**

Normal Price £372.35

Special Price £347.40 (inc VAT)
BEEBUG Members only.

Send to: Z88 Offer, BEEBUG Ltd,
Dolphin Place, Holywell Hill, St Albans, Herts
AL1 1EX.
or Phone (0727) 40303.

This special offer is available for a limited period only.
Please see our Retail Catalogue for price and availability of
the Z88 and individual peripherals.



Title Spitfire '40
Supplier Mirrorsoft
Price £9.95 (tape)
 £12.95 (5.25" disc)
 £14.95 (3.5" disc)

Rating ****

As the name suggests, Spitfire '40 puts you at the controls of this famous aeroplane, as a newly trained

pilot posted to a Spitfire Squadron in South East England during Wartime.

In many respects this game resembles the famous Aviator from Acornsoft, except that Spitfire '40 records your experience from one game to another, and allows you to move through the ranks, and be awarded medals for daring deeds. The aircraft can be controlled with or without joysticks, but I would not recommend using the keyboard because of the



Title The Life Of Repton
Supplier Superior Software
Price £6.95 (tape)
 £7.95 (5.25" disc)
 £9.95 (3.5" disc)

Rating ***

First there was Repton, then there was Repton 2 and Repton 3, and finally through popular demand Superior Software has

released The Life Of Repton. This package is much like Repton 3 but has forty more action-packed screens depicting Repton's life from when he was but a small toddler surrounded by teddy bears to an old age pensioner with false teeth and spectacles.

For those who have managed to escape the Repton bug until now, this game offers great graphics and sound effects, along with many taxing puzzles. You control Repton as he roves around the landscape searching for goodies



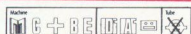
Title The Living Daylights
Supplier Domark
Price £9.95 (tape)
 £12.95 (5.25" disc)

Rating **

Domark seem to have captured a niche in the computer games market by designing games around films and books that are

already well known. In The Living Daylights you play the famous James Bond 007. The game takes you through all the major scenes of the film. In true James Bond tradition the game consists of shooting the enemy, defending the helpless government agent, and winning the favours of the beautiful woman.

The adventure starts in Gibraltar where Bond finds himself face to face with the SAS. Then on to the Lenin People's Music Conservatory, where Bond must rescue the Russian defector



Title Spellbinder
Supplier Superior Software
Price £9.95 (tape)
 £11.95 (5.25" disc)
 £14.95 (3.5" disc)

Rating ***

Spellbinder is a very interesting three dimensional adventure game. You play the part of a Wizard in a castle full of nasties, tricks,

and puzzles. Your object is to destroy the evil Wizard Zorn with sorcery. The Wizard is easily directed around the castle in search of the necessary ingredients to mix a magic potion. This will enable you to cast the spell and destroy the evil Zorn. Unfortunately, there are a large number of bad characters wandering around the castle as well. You have just a few spells at your disposal, providing that you can find and mix the appropriate ingredients.

Extensive use is made of the function keys, and

large number of controls necessary to manoeuvre the plane and chase enemy aircraft effectively.

There are three modes of play depending upon your flying experience. You can either practice taking off and landing, or gain some combat experience 10,000 feet up in the clouds. For the actual combat playing mode, you are given a specific mission. You then have to get the plane airborne and locate the enemy aircraft.

such as teddy bears and records. Unfortunately the area is full of hazards such as office equipment, aggressive clockwork toy soldiers, and endless cups of coffee! These hazards must be avoided at all costs, which is not always so easy.

If you are already a dedicated Repton fan, then you will definitely want to add this to your collection. If you have not yet played any of the Repton games but feel tempted, then this is by far the best Repton game to buy. **B**

Koskov and smuggle him to safety. Your mission takes you to the rooftops of Tangiers, and then into the middle of a Russian Air Base. Survive the final confrontation with the evil Brad Whittaker, and the beautiful Kara is Bond's.

Unfortunately, although it has a brilliant plot to follow and is nicely presented, I found the game quite unexciting. The first few screens remind me very much of the old fairground shooting arcades that involved firing at objects

a system of icons allows the Wizard to select spells, move around, and examine objects easily and quickly. Although the game is very addictive, I was disappointed to find that all the graphics are in monochrome. The graphics are detailed, but some effect is lost because the characters tend to blend in with the background. I do not feel that this game is really up to Superior's usual high standard. Having said that, it kept me amused for some time, and I would recommend it to any adventure game enthusiast. **B**

Once you have destroyed the enemy, you must return to base and land without crashing.

All in all, I thoroughly enjoyed playing this game, and it kept me occupied for hours. There are some great visual and sound effects, and the instruction manual is well written and presented. Providing that you have some joysticks, I would thoroughly recommend Spitfire '40 as a stocking filler this Christmas. **B**



as they appear and disappear in the distance. The graphics are not particularly smooth, and Bond is a little difficult to control. This is another game that is better played with a joystick.

I was disappointed with this game, and would have expected better quality graphics and sound, considering the large amount of publicity hype that this game has been given. If you are looking for a good game this Christmas I would look elsewhere. **B**



WORD-EX AND WORDWISE PLUS II

Dave Somers evaluates two new products, Word-Ex and Wordwise Plus II, designed to enhance Wordwise Plus usage.

INTRODUCTION

Although Wordwise and Wordwise Plus are fine word processors, they do tend to look a bit outdated when compared to the facilities that are available with the likes of InterWord. The Word-Ex ROM from Norwich Computer Services enables such facilities to be incorporated into Wordwise or Wordwise Plus. It operates as an extension to Wordwise, offering many utilities and functions.

An alternative upgrade path is offered by IFEL, who in co-operation with Computer Concepts have produced a complete replacement ROM for Wordwise, the new version being called Wordwise Plus II.

Both products offer an upgrade route for Wordwise and Wordwise Plus users. This review looks at what both products have to offer.

WORD-EX

Word-Ex consists of a 16K EPROM, a 32 page A5 instruction manual, and a reference card. The ROM may be fitted into any vacant sideways ROM socket. Word-Ex claims three pages of memory for workspace. If you are not using Word-Ex at any time, then this memory can be released. This will not be necessary on the Master or Compact as the memory claimed will be in private RAM, and not user RAM.

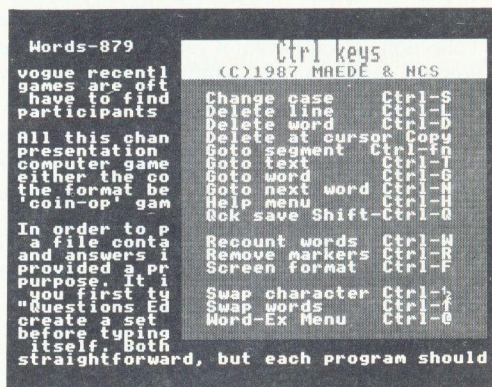
Product	Word-Ex
Supplier	Norwich Computer Services Freepost, 18 Mile End Road, Norwich NR4 7BR. Tel. (0603) 507057
Price	£30.00 inc. VAT and p&p

To activate Word-Ex, the command *EWORDWISE should be entered. This will select Wordwise (Plus) as the current language, and enable Word-Ex. Alternatively, if one is already in the Wordwise (Plus) environment, the command *EON should be used.

FILE ACCESS

When loading or saving text files, Word-Ex makes the whole process easier by displaying a list of files on the screen. Any of these may be loaded by moving a cursor to its position on the screen, or by typing in its name.

One extremely useful function is the "quick save". When editing text, pressing Shift-Ctrl-Q will cause the current text to be saved to disc. The filename to be used should be stored in the first line of your text file.



EDITING FACILITIES

The editing facilities provided by Word-Ex are accessed by a series of Ctrl key combinations. A fast find routine ("Goto") enables you to go direct to any piece of selected text.

One of the most common typing errors is that of transposition of characters. Word-Ex can swap two adjacent characters (Ctrl-V) or two adjacent words (Ctrl-^), to correct such typing errors. Commands such as delete line (Ctrl-L), and delete at character (Copy), that are found on InterWord, are now provided thanks to Word-Ex.

THE WORD-EX MENU

The remainder of Word-Ex's facilities are accessed by pressing Ctrl-@ which produces a further selection menu. One option allows paragraphs to be sorted either alphabetically or numerically, in either ascending or descending order. This sort can be over the whole of the text, or restricted to a specific marked section.

The 'case change' function allows the case of the text to be changed in a more convenient manner than does the Ctrl-S 'swap case' facility. There are also options to change the text so that words may be forced to lower case or upper case, 'all set as titles', and set 'intelligently'. For the "All Set As Titles" option, words have their initial letter converted to upper case. For "Intelligent" case change, all text is converted to lower case, except for letters at the beginning of new sentences, which are changed to upper case.

MISCELLANEOUS FACILITIES

This option offers a variety of additional functions. To enable the red function keys to be programmed easily, a function key editor is provided. When you have selected which key to edit, you simply press the key actions desired. You don't, for example, have to remember that "!!" is cursor down, you simply press the cursor down key, etc. Simplicity itself!

THE STAR COMMANDS

Word-Ex comes equipped with a host of star commands - twenty-seven of them. These are used to invoke Word-Ex's various functions without having to go through its drop-down menus, and also to use them from within segment programs. They can of course be programmed into the function keys to allow repeated operations to be executed at the press of a key.

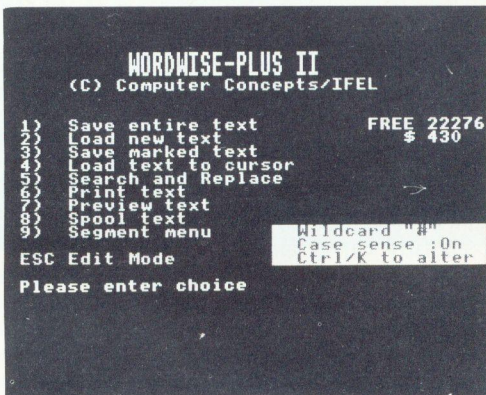
All-in-all, Word-Ex has much to offer the Wordwise enthusiast looking for more features, and the star commands are an added bonus.

WORDWISE PLUS II

Unlike the Word-Ex ROM, Wordwise Plus II is a re-write of Wordwise Plus (with Computer Concepts' blessing), and not an extension ROM for Wordwise (Plus). The software is supplied

as a 32K ROM mounted on a small carrier board for insertion in a single sideways ROM socket, and works on all BBC micros (except, of course, the Archimedes).

Product Supplier	Wordwise Plus II IFEL, 36 Uplands Drive, Plymouth, Devon PL6 6BD. Tel. (07555) 7286
Price	£66.95 inc. VAT and p&p £40.95 inc. (upgrade from Wordwise Plus)



A 102 page A5 instruction manual details the additional facilities offered by Wordwise Plus II, and an accompanying disc contains various segment programs, and other utilities. You will, however, need to refer to the original Wordwise Plus manual if you are new to Wordwise Plus, and this is supplied as part of the package.

The only immediate difference from Wordwise is the presence of an "information window" showing the current availability of free memory, current wildcard setting, and whether searches are case sensitive or not.

FILE HANDLING

Loading and saving of text can be performed using the familiar options 1 and 2 from the main menu. If, however, Shift-1 or Shift-2 is pressed, an InterWord style menu for selecting a filename is produced, making the whole operation much easier.

The system is quite sophisticated and copes with any Acorn compatible DFS and ADFS systems. It will also cope with Watford 62-file DFS discs, and even Solidisk multiple catalogue DFS discs.

CONTROL KEYS

When editing text, a host of additional Ctrl-key facilities is available. You can set markers around a single word, or paragraph, and remove it completely. If you then press Ctrl-M a pop-up menu appears so that you can choose to move, delete, or copy the marked section of text.

Setting up the a layout is now simply a matter of pressing Ctrl-O. A pop-up menu is displayed and you can select the options required. Even the text for headers and footers can be entered from here. It's a lot easier to remember than all those embedded commands.

PRINTER CONFIGURATION

Similarly, Ctrl-E selects the "Enter Print Codes" menu, allowing easy selection of different printer styles. When the desired option has been selected, the correct embedded command is automatically entered into the text. To configure Wordwise Plus II for your printer, the Printer Definer menu can be used with Ctrl-I. It comes complete with the various RPS (Redefine Printer Sequence) codes for Epson or Brother printers, or you can load them from a disc file which you have to create.

SEARCH AND REPLACE

With Wordwise Plus you can only access the search and replace facility from the main menu. With Wordwise Plus II in edit mode, if you press Ctrl-X, a pop-up menu appears for you to select the search and replace strings. It operates in the same manner as the search and replace in InterWord, in that you can then select case sensitivity, and global or selective operation.

A "Find String" facility can be called upon with Ctrl-G (for Goto). A small pop-up window appears and the text to be searched for is entered there. After the first find of the text, Ctrl-Q will then take the cursor to subsequent occurrences of the text.

CHANGING CASE

In addition to the familiar Ctrl-S to swap between upper and lower case, two new case change functions are provided. Pressing Ctrl-U forces the character under the cursor into upper case, whereas Ctrl-Y forces the text to lower case. Character transposition errors can also be corrected by pressing Ctrl-@.

UTILITIES DISC

A utilities disc is supplied with the package containing a number of segment programs, and other useful items. One of the more useful programs will compact a Wordwise Plus segment program. This has two advantages; the program will be shorter, and it will execute faster. Another useful program on the disc will strip out all the embedded commands from a piece of text, which could be useful when transferring text to another word processor.

THE BASIC EDITOR

This is similar in operation to the EDIT function provided by Basic IV on the Master, where a Basic program is de-tokenised and placed in the Master's Editor. In this case, a small machine code program is loaded, and when f0 is pressed, any resident Basic program is de-tokenised and loaded into Wordwise Plus II. When you've finished editing the program, it can be re-tokenised and returned to Basic.

CONCLUSIONS

If you wish to upgrade your Wordwise or Wordwise Plus, you could go for one of the above ROMs, or change to InterWord. The choice is up to you. If you like the Wordwise style and programming language, upgrade with either Word-Ex or Wordwise Plus II, otherwise, particularly if you want a WYSIWYG style edit screen choose InterWord.

Of the two packages, I felt that Wordwise Plus II was the better. The pop-up windows were nicely presented, and it performed very well. It has the feel of InterWord, yet the style of Wordwise, making a great combination.

Word-Ex, though, offers all of those utility star commands, and if you need to use such facilities often, then undoubtedly these would be useful.

B

TRIVIA Quiz

Jonathan Temple presents Trivia Quiz, a game for all the family this Christmas.

The board game 'Trivial Pursuit' has been an immense success, and trivia quiz games in general have been much in vogue recently. Unfortunately, such games are often quite expensive, and you have to find several other willing participants in order to play.

All this changes with this month's presentation of the BEEBUG Trivia Quiz computer game! You can play against either the computer or another player, the format being similar to the 'coin-op' games now common in pubs.

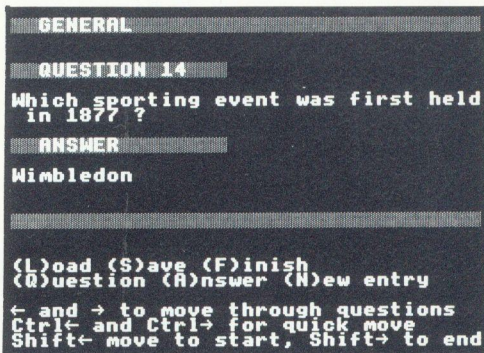
In order to play the BEEBUG Trivia Quiz, a file containing a set of questions and answers is required. We have provided a program for just this purpose. It is suggested therefore, that you first type in and save the "Questions Editor" program, using it to create a set of questions and answers before typing in the Trivia Quiz program itself. Both programs are quite straightforward, but each program should be treated separately and be typed in and saved before running.

PLAYING THE BEEBUG TRIVIA QUIZ

With the BEEBUG Trivia Quiz program you can choose to play against either the computer, or against another (human) opponent. Next, a set of question files (in the Q directory) is displayed (not on a cassette system), and you will be asked to enter the name of the required subject file. This will then be loaded in from disc or tape. These files can be created using the "creation" program described later. A ready-made set of questions and answers is to be found on this month's magazine disc/tape.

A noughts-and-crosses board is displayed on the screen, and you must attempt to complete a line of your counters in the normal way. You do this by selecting a square, after which you must

answer a question correctly before the counter is placed. Answer it incorrectly and your opponent's counter is placed there instead, the object being to complete a line before your opponent. If no player completes a line of counters, the player with the most counters wins the game. At the end of the game you can choose a new game, and choose the same or a different subject on which to be questioned.



CREATING QUESTIONS

When run, the second program allows you to create and edit files of up to 200 questions. To enter a new question, press N (for New question), and the cursor will move to the question line where you may then type in a question up to 70 characters long. On pressing Return, the cursor moves to the answer line which may be up to 16 characters long. Existing entries can be edited by pressing Q (edit question) or A (edit answer).

To make entry and editing easier, a simple line editor is supplied. The left and right cursor keys move along the line, Ctrl-A deletes the character at the cursor and the Delete key deletes to the left of the cursor. Characters typed are inserted at the current cursor position. Press Return when you are satisfied.

The left and right cursor keys move to and through the list of questions. Pressing Ctrl at the same time skips questions ten at a time, while pressing Shift at the same time moves you to either the first or last question, Wordwise style. Individual questions can be deleted by using the Delete key, while the whole set of questions can be deleted by pressing f0. In both cases a check is made first.

Question files can be saved and loaded using keys 'S' and 'L'. You must supply a filename which should be a maximum of 7 characters for DFS, 10 characters for ADFS and 8 characters for cassette. When a file is saved, its name is automatically prefixed with "Q." (disc users will find question files in directory Q). The resulting file can then be loaded into the main Trivia Quiz program.

Users of the ADFS should either set up a directory Q or delete the references to the "Q." prefix in the programs. In the latter case, alter line 1820 in the Quiz program, and line 1730 in the Questions Editor program.

A useful tip when creating questions and answers is to make sure that the answers are consistent in their form. For instance, when the questions from the magazine disc/tape files prompt for a person's name, their whole name must be typed. When they ask for a number, this should be entered as "9" rather than "nine" for instance.

The Trivia Quiz program is quite free in the answers it accepts. There is no distinction between cases, upper or lower, and it will accept most mis-spellings, since it ignores vowels and the more redundant consonants. Sometimes this may result in a wrong answer being marked right (or vice versa) but this is rare. A consequence of the acceptance of mis-spellings, is that when creating question files should you find that an answer is longer than 16 characters, you can shorten it by missing out spaces and vowels.

```

10 REM Program Trivia Quiz
20 REM Version B1.6
30 REM Author Jonathan Temple
40 REM BEEBUG December 1987
50 REM Program subject to copyright
60 :
100 MODE 7:ON ERROR GOTO 2670
110 S%=&3700:HIMEM=S%:PROCcode:PROCinit
130 REPEAT:PROCplayers
140 REPEAT:CLS:PROCload
160 REPEAT:R=FNplay:UNTIL R>1
190 UNTIL R>2:UNTIL R<>3:VDU26,12
220 END
230 :
1000 DEFPROCplayers
1010 VDU 26,12,31,0,4

```

```

1020 PRINT"Number of players (1/2) : ";
1030 REPEAT:K=GET AND &CF
1040 UNTIL K=1 OR K=2
1050 NP=3:PL=1:PRINT K
1060 IF K=1 THEN N$(2)="The computer":N
P=0
1070 FOR N=1 TO K
1080 PRINT""Please enter your name, pl
ayer ";N""-->";
1090 N$(N)=FNinput(13)
1100 NEXT
1110 S(1)=0:S(2)=0
1120 ENDPROC
1130 :
1140 DEFNplay
1150 PROCscreen :PROCrandom
1160 REPEAT
1170 PL=PL EOR NP :PROCmove(PL)
1180 UNTIL W>0
1190 VDU 28,3,24,36,15,12,26
1200 PROCdb(0,15,CHR$(137-W*3)+N$(W)+"
wins...!")
1210 S(W)=S(W)+1
1220 IF W<3 PRINTTAB(2,18);CHR$134;N$(1
);" ";S(1);" ";N$(2);" ";S(2)
1230 IF FNYes("Same game") THEN =1
1240 IF FNYes("New questions") THEN =2
1250 IF FNYes("Restart") THEN =3
1260 =4
1270 :
1280 DEFNyes(Q$)
1290 PRINTTAB(3,21)SPC(20);
1300 PRINTTAB(3,21)Q$;" (Y/N) : ";
1310 REPEAT K=GET AND &DF
1320 UNTIL K=78 OR K=89
1330 =(K=89)
1340 :
1350 DEFPROCmove(PL)
1360 VDU 28,3,20,38,15,12
1370 PRINT"Your move, ";CHR$(137-PL*3);N
$(PL);" : ";
1380 REPEAT K=GET AND 159
1390 SOUND 18,-8,50,2
1400 UNTIL K>0 AND K<10
1410 IF B(K)>0 GOTO 1360
1420 VDU K+64,28,3,24,36,18,12
1430 N=R%?T:T=T+1
1440 W=0:IF T=10 THEN W=3
1450 PROctext($FNq(N))
1460 PRINT""-->"; :$U%=$FNa(N)
1470 CALL check :C$=$U%
1480 C$=$U%
1490 $U%=$FNinput(25)
1500 CALL check :A$=$U%
1510 A$=$U%
1520 IF LEN(C$)>LEN(A$) C=0 ELSE C=INST
R(A$,C$)
1530 CLS :IF C>0 THEN PROCdb(0,0,"CORRE

```



```

CT"):SOUND1,1,100,3 ELSE PROCdb(0,0,"WRO
NG"):SOUND0,-12,6,2:PL=PL EOR 3
1540 VDU 26:B(K)=PL
1550 PROCsquare(K) :W=FNline(PL)
1560 TIME=0:REPEAT UNTIL TIME>200
1570 ENDPROC
1580 :
1590 DEFFNline(C)
1600 RESTORE
1610 FOR N=1 TO 8:READ S1,S2,S3
1620 IF B(S1)=C IF B(S2)=C IF B(S3)=C T
HEN W=C
1630 NEXT
1640 =W
1650 :
1660 DATA 1,2,3,4,5,6,7,8,9
1670 DATA 1,4,7,2,5,8,3,6,9
1680 DATA 1,5,9,3,5,7
1690 :
1700 DEFPROCrandom
1710 FOR N%=1 TO Q% :R%=R%+N%:NEXT
1720 FOR N%=1 TO 10 :M%=R%+RND(Q%-1)+1
1730 A%=R%?N%:R%?N%=M%?:M%=A%
1740 NEXT
1750 T=1
1760 ENDPROC
1770 :
1780 DEFPROCload
1790 PRINT""""Current file - ";CF$
1800 PRINT""Type filename:";
1810 F$=FNinput(8)
1820 $%700="L. Q."+F$+" "+STR$(S%)
1830 X%=0:Y%=&7:CALL &FFF7
1840 Q%=?S%:IF Q%>200 T=1 :ENDPROC
1850 CF$=F$
1860 ENDPROC
1870 :
1880 DEFFNinput(M)
1890 LOCAL K,L :L$="":VDU 134
1900 REPEAT L=LEN(L$):K=GET
1910 IF K=127 IF L>0 L$=LEFT$(L$,L-1):G
OTO 1940
1920 IF K=13 IF L>0 GOTO 1940
1930 IF K>31 AND K<127 IF L<M L$=L$+CHR
$(K) ELSE K=7
1940 VDU K
1950 UNTIL K=13
1960 =L$
1970 :
1980 DEFPROCscreen
1990 VDU 26,12
2000 PROCdb(19,4,"T R I V I A")
2010 PROCdb(21,8,"Q U I Z")
2020 FOR N%=1 TO 14:VDU31,0,N%,154:NEXT
2030 FOR N=1 TO 9:B(N)=0
2040 PROCsquare(N)
2050 X=(N-1) MOD3:Y=(N-1) DIV3
2060 VDU 31,X*5+4,Y*4+3,64+N

```

```

2070 NEXT
2080 ENDPROC
2090 :
2100 DEFPROCsquare(N)
2110 X=(N-1) MOD3:Y=(N-1) DIV3
2120 VDU 31,X*5+2,Y*4+2:PRINT C$(B(N))
2130 ENDPROC
2140 :
2150 DEFPROCtext(T$)
2160 REPEAT
2170 F$=LEFT$(T$,33):T$=MID$(T$,34)
2180 IFRIGHT$(F$,1)=" " OR T$="" GOTO 2
230
2190 REPEAT T$=RIGHT$(F$,1)+T$
2200 F$=LEFT$(F$,LEN(F$)-1)
2210 UNTIL RIGHT$(F$,1)=" " OR F$=""
2220 IF F$="" PRINT
2230 PRINT F$:UNTIL T$=""
2240 ENDPROC
2250 :
2260 DEFFNq(N%)=V%+N%*88
2270 :
2280 DEFFNa(N%)=W%+N%*88
2290 :
2300 DEFPROCdb(X%,Y%,T$)
2310 FOR N%=Y% TO Y%+1
2320 VDU 31,X%,N%,141,135
2330 PRINT T$;
2340 NEXT
2350 ENDPROC
2360 :
2370 DEFPROCinit
2380 DIM B(9),S(3),N$(3),C$(2)
2390 DIM R% 200
2400 VDU 23;10,32;0;0;0;
2410 V%=S%-87:W%=V%+71
2420 N$(0)="" :N$(3)="Nobody"
2430 CF$="None"
2440 ENVELOPE 1,133,8,4,8,3,1,1,80,0,0,
-10,80,0
2450 Y$=CHR$147:C$=CHR$150:W$=CHR$148
2460 R$=CHR$135+CHR$10+CHR$8+CHR$8+CHR$
8+CHR$8+CHR$8
2470 C$(0)=W$+"<,1"+R$+W$+"5 j"+R$+W$+"
-,""
2480 C$(1)=C$+"t x"+R$+C$+"b"+CHR$255+"
1"+R$+C$+" " "+"
2490 C$(2)=Y$+"x,t"+R$+Y$+"5 j"+R$+Y$+"
+,""
2500 ENDPROC
2510 :
2520 DEFPROCcode
2530 RESTORE 2620:C%=0
2540 FOR M%=&5700 TO &572C STEP 4
2550 READ A$ :M%=EVAL("&"+A$)
2560 C%=C%+!M%:NEXT
2570 IF C%<&DEF7E72D PRINT'"Error in P
ROCcode or DATA lines":END

```



```

2580 check=&5700:U%=check+80
2590 $(check+64)=" aeioyuh."
2600 ENDPROC
2610 :
2620 DATA 708600A2,C95750BD,C9069041
2630 DATA 0902B05B,F071C520,A0718513
2640 DATA 5740D908,10880AF0,9970A4F8
2650 DATA 70E65750,D00DC9E8,60D7
2660 :
2670 IF ERR>128 PRINT:REPORT:PRINT" err
or";:TIME=0:REPEAT UNTIL TIME>250:GOTO12
0
2680 MODE 7
2690 PRINT':REPORT
2700 PRINT " at line ";ERL
2710 END

```

```

10 REM Program Questions Editor
20 REM Version B2.1
30 REM Author Jonathan Temple
40 REM BEEBUG December 1987
50 REM Program subject to copyright
60 :
100 REM Find current filing system
110 A%=0:Y%=0:fs=USR(&FFDA) AND &FF
120 DFS=(fs=4):ADFS=(fs=8):CFS=(fs=1)
130 MODE 7:ON ERROR GOTO 2690
140 S%=&3700:HIMEM=S%
150 PROCinit:PROCclear:PROCsetup
160 PROCedit:VDU 26,12
170 PROCon:END
180 :
1000 DEFPROCedit
1010 PRINTTAB(1,17)"(L)oad (S)ave (F)in
ish"
1020 PRINTTAB(1,18)"(Q)uestion (A)nswer
(N)ew entry"
1030 PRINTTAB(1,20)"[ and ] to move thr
ough questions"
1040 PRINTTAB(1,21)"Ctrl[ and Ctrl] for
quick move"
1050 PRINTTAB(1,22)"Shift[ move to star
t, Shift] to end"
1060 PROCtitles:N%=1:Q%=?S%+1
1070 REPEAT
1080 REPEAT
1090 PROCdisplay(N%)
1100 REPEAT
1110 IF INKEY(-122) PROCforward
1120 IF INKEY(-26) PROCbackward
1130 IF INKEY(-86) PROCcenter
1140 IF INKEY(-17) PROCeditq
1150 IF INKEY(-66) PROCedita
1160 IF INKEY(-90) PROCdelete
1170 IF INKEY(-87) PROCfile("LOAD")
1180 IF INKEY(-82) PROCfile("SAVE")
1190 TIME=0:REPEAT UNTIL TIME>6
1200 UNTIL INKEY(-33) OR INKEY(-68)

```

```

1210 IF INKEY(-33) PROCclearall
1220 UNTIL INKEY(-68)
1230 UNTIL FNsured("QUIT")<>0
1240 ENDPROC
1250 :
1260 DEFPROCforward
1270 IF INKEY(-2) IF N%+10<Q% N%=N%+10:
GOTO 1300
1280 IF INKEY(-1) IF Q%>1 IF N%<>Q%-1 N
%=Q%-1:GOTO 1300
1290 IF N%>Q%-2 ENDPROC ELSE N%=N%+1
1300 PROCdisplay(N%)
1310 ENDPROC
1320 :
1330 DEFPROCbackward
1340 IF INKEY(-2) IF N%>10 N%=N%-10:GOT
O 1370
1350 IF INKEY(-1) IF N%>1 N%=1:GOTO1370
1360 IF N%=1 ENDPROC ELSE N%=N%-1
1370 PROCdisplay(N%)
1380 ENDPROC
1390 :
1400 DEFPROCcenter:
1410 IF Q%=201 VDU7:ENDPROC
1420 PRINTTAB(0,2);SPC(MQ);TAB(0,7);SPC
(MA);TAB(11,0);Q%;TAB(0,2);
1430 N%=Q%:Q%=Q%+1
1440 $FNq(N%)=FNedit("",MQ)
1450 PRINT TAB(0,7);
1460 $FNa(N%)=FNedit("",MA)
1470 ENDPROC
1480 :
1490 DEFPROCeditq:PRINT TAB(0,2);
1500 $FNq(N%)=FNedit($FNq(N%),MQ)
1510 ENDPROC
1520 :
1530 DEFPROCedita:PRINT TAB(0,7);
1540 $FNa(N%)=FNedit($FNa(N%),MA)
1550 ENDPROC
1560 :
1570 DEFPROCdelete:IF Q%=1 ENDPROC
1580 IF FNsured("Delete")=0 ENDPROC
1590 $FNq(N%)="":$FNa(N%)=""
1600 IF N%=Q%-1 IF N%>1 N%=N%-1:GOTO164
0
1610 PROCb(0,10,36,"Please wait")
1620 FOR M%=FNq(N%) TO FNq(Q%) STEP4
1630 !M%=M%!U%:NEXT
1640 PROCdisplay(N%):Q%=Q%-1
1650 PROCb(0,10,36,STRINGS(34," "))
1660 ENDPROC
1670 :
1680 DEFPROCfile(A%):LOCAL L
1685 IF A%="SAVE" IF Q%<11 VDU7:ENDPROC
1690 IF FNsured(A%)=0 ENDPROC
1700 CLS:PROCb(0,10,21,"FILENAME: ")
1710 IF ADFS L=10 ELSE IF CFS L=8 ELSE
L=7

```



```

1720 F$=FNinput(L):?S%=Q%-1
1730 $&700=A$+" Q."+F$+" "+STR$(S%)
1740 IF A$="SAVE" $&700=$&700+" "+STR$(
(FNq(Q%)) ELSE N%=1
1750 X%=0:Y%=&7:CALL &FFF7
1760 Q%=?S%+1:PROctitles
1770 PROCdisplay(N%)
1780 ENDPROC
1790 :
1800 DEFFNsured(T$)
1810 VDU31,1,10:*FX 21
1820 PROCb(0,10,36,T$+": Are you sure (
Y/N) ? ")
1830 REPEAT G%=GET AND 223
1840 UNTIL G%=78 OR G%=89
1850 PROCb(0,10,36,STRING$(34," "))
1860 =(G%=89)
1870 :
1880 DEFPROCtitles
1890 VDU28,1,16,38,0,12,26
1900 PROCb(1,1,36,F$)
1910 PROCb(1,4,17,"QUESTION")
1920 PROCb(1,9,17,"ANSWER")
1930 PROCb(1,14,36,"") :VDU28,1,16,35,4
1940 VDU 28,1,16,35,4
1950 ENDPROC
1960 :
1970 DEFPROCb(X%,Y%,T%,T$)
1980 VDU31,X%,Y%,157,135,31,T%,Y%,156
1990 PRINT TAB(X%+2,Y%) T$:
2000 ENDPROC
2010 :
2020 DEFPROCdisplay(N%)
2030 PRINTTAB(11,0);N%," ";
2040 PRINTTAB(0,2) SPC(MQ);TAB(0,2);$FN
q(N%);TAB(0,7);SPC(MA);TAB(0,7);$FNa(N%)
2050 ENDPROC
2060 :
2070 DEFFNinput(M%)
2080 PROCCon:L$="":*FX 21
2090 REPEAT L%=LEN(L$)
2100 A%=GET
2110 IF A%=127 IF L%>0 L$=LEFT$(L$,L%-1
):GOTO2140
2120 IF A%=13 IF L%>0 GOTO2140
2130 IF A%>31 AND A%<127 IF L%<M% L$=L$
+CHR$(A%) ELSE A%=7
2140 VDU A%
2150 UNTIL A%=13
2160 PROCoff
2170 =L$
2180 :
2190 DEFFNedit(E$,M%)
2200 PROCCon:*FX 21
2210 A%=POS:B%=VPOS
2220 PRINT E$;TAB(A%,B%);
2230 L%=1:*FX 4,1
2240 REPEAT G%=GET

```

```

2250 IFG%=136 IF L%>1 VDU8:L%=L%-1
2260 IFG%=137 IF L%<=LEN(E$) VDU9:L%=L%
+1
2270 IFG%=127 IF L%>1 IF LEN(E$)>0 PROC
add(-2,0,"",8,-1)
2280 IF G%=1 IF LEN(E$)>1 PROCadd(-1,1,
"",0,0)
2290 IF G%>31 IF G%<127 IF LEN(E$)<M% P
ROCadd(-1,0,CHR$(G%,9,1)
2300 UNTIL G%=13
2310 PROCoff:*FX 4
2320 =E$
2330 :
2340 DEFPROCadd(A,B,G$,V,N)
2350 E$=LEFT$(E$,L%+A)+G$+MID$(E$,L%+B)
2360 X=POS:Y=VPOS:L%=L%+N
2370 PRINTTAB(A%,B%);E$;" ";TAB(X,Y);CH
R$(V);
2380 ENDPROC
2390 :
2400 DEFFNq(N%)=V%+N%*U%
2410 :
2420 DEFFNa(N%)=W%+N%*U%
2430 :
2440 DEFPROCclearall
2450 IF FNsured("CLEAR") THEN PROCclear
2460 ENDPROC
2470 :
2480 DEFPROCclear
2490 N%=1:Q%=1:$FNq(1)=""$:FNa(1)=""
2500 ?S%=0:ENDPROC
2510 :
2520 DEFPROCCon:VDU23;10,16;0;0;0;0;
2530 ENDPROC
2540 :
2550 DEFPROCoff:VDU23;10,32;0;0;0;0;
2560 ENDPROC
2570 :
2580 DEFPROCinit
2590 MQ=70:MA=16:U%=MQ+MA+2
2600 V%=S%+1-U%:W%=V%+MQ+1
2610 ENDPROC
2620 :
2630 DEFPROCsetup
2640 F$="NO FILE":PROCoff
2650 VDU12,31,0,1,132,31,0,4,132
2660 VDU31,0,9,132,31,0,14,132
2670 ENDPROC
2680 :
2690 VDU 26:*FX 4
2700 CLS
2710 IF ERR>127 VDU26,12:PRINT':REPORT:
PRINT'"Press any key":IF GET GOTO 150
2720 ON ERROR OFF
2730 PRINT':REPORT
2740 PRINT" at line ";ERL'
2750 PROCCon:END

```


HINTS HINTS HINTS HINTS HINTS

and tips and tips and tips and tips and tips

EASIER GAME PLAY

Nicholas Sayers

After many hours of playing *Imagin's Arkanoid*, I have come up with a pretty nifty little 'cheat' method: simply hold down the Delete and Copy keys while playing the game, and you should find that the Vaus craft slows down to half its usual speed. This works because the game is continually pausing and unpausing. This cheat method should also work for other games which provide you with the means to pause, and re-continue with the game.

POWER ON RESET

Dave Somers

Every now and then you might find that you need to turn your machine off and then on again - possibly because a program (usually machine code) has gone wrong. To save wear and tear on the power supply unit on/off switch (and the computer), if you can still type anything, just try:

```
*FX151,78,127
```

and then press Break. This will fool the computer into thinking it has just been switched on, and it will then perform a cold reset. This is also useful if you want a program to exit by clearing all of memory. The computer will automatically reset itself if the program issues a CALL !-4 on exit immediately after using the above FX call.

GETTING INPUT

Jonathan Temple

When you want to input a character and ensure that the resulting ASCII code is the upper case equivalent of the key pressed, use:

```
G=GET AND &DF
```

Thus, if you press "x", the value in G will be the ASCII value of "X". Moreover, if you use:

```
G=GET AND &CF
```

when pressing any number key (0 to 9), the resulting code will be converted to the key's numeric value. So pressing the "2", for example, key will set the subsequent value of G to be 2.

SAVING MEMORY WHEN USING DATA STATEMENTS

Robert Alcock

Have you ever run out of memory when using long numerical data statements (e.g. when plotting graphics points)? Usually, in such cases, there is a zero at the end of each number. To save memory, place such numbers in the data statements with the last zero missing (i.e. divided by 10) and then multiply by 10 after reading in each value to restore it to its correct value. So, if you had the following data:

```
DATA 500,840,420,790,...
```

these would be re-written as:

```
DATA 50,84,42,79,...
```

The data could be read in and restored to its correct form using:

```
READ A
```

```
IF A<=128 A=A*10
```

Please note that this only works for numbers ranging from 100 to 1280. Numbers below 100 (e.g. 42) would have to be stored as 4.2, and numbers above 1280 would have to be written fully. You could, of course, change the upper limit to suit your own requirements.

WHICH MODE?

Jonathan Wilkinson

Unlike other languages such as Comal, Basic provides no keyword or variable which allows a program to determine the current screen mode. However, you can use the following short function to return the value of the current mode by calling the appropriate machine code routine. Please note that the value is always in the range of 0 to 7 regardless of whether shadow RAM is active.

```
10 DEF FNmode
20 A%=&87
30 M%=USR(&FFF4)
40 M%=M% AND &FF0000
50 M%=M% DIV &10000
60 =M%
```

The routine uses OSWORD &87 to read the screen mode via the USR keyword in line 30. Lines 40 and 50 strip out the other data returned by this call in order to return the mode at line 60.

B



POSTBAG



POSTBAG

PROBLEMS WITH PROGRAMS

As a BBC computer user, but computer novice, my recent experience with Paint & Draw (BEEBUG Vol.6 No.4) may be of interest to others. Despite the caveat in the article about the Watford DFS system, the program was successfully demonstrated with a Watford DFS in the BEEBUG shop, so I bought a copy of the disc. Guess what? It didn't work when I got it home.

A phone call to BEEBUG established that I had an early version of the Watford DFS, so a quick trip down the road to Watford saw the purchase of the Watford DFS version 1.44 (cost £5.75). But still a blank screen faced me, and my children nearly learnt some choice new vocabulary.

I eventually realised that the only other non-standard item in my machine was a Watford 32K RAM card. Disabling this, everything then worked. Can anyone explain why the Watford board has this effect?

D.R.Hall

Mr Hall raises several points. All programs published in BEEBUG are tested on standard Acorn systems (model B, B+, Master etc). Unfortunately, some early versions of non-Acorn DFS systems contained bugs (as with Watford's) and it is advisable to upgrade if possible at the relatively small cost involved.

*The majority of programs that save and load screens (as does Paint & Draw) use *SAVE and *LOAD for this purpose, addressing the normal screen memory area. If any shadow RAM (added on to a model B, or integral as with the Master or Compact) is enabled, then the screen save and load will be to the wrong area of memory. This is a point well worth remembering as it can otherwise cause unexpected results. For example, loading text into Wordwise on a Master results in a blank screen if shadow RAM is switched on.*

ONE MORE GIANT BOUNCE

If you make the following alterations to the program, Giant Bounce (BEEBUG Vol.6 No.5) it will work just as well with Basic I as it does with the listed Basic II version. Delete line 630 and then add:

```
90 DIM screendata 16
95 !screendata=&888888FF
:screendata!4=&FF888888:scr
eendata!8=&111111FF:screend
ata!12=&FF111111
```

In the second program delete lines 900 to 950 and then add:

```
90 DIM sound 8
212 !sound=&FFF50001
214 sound!4=&00010001
```

Robert Alcock

BEEBUG members often respond to magazine programs with suggested improvements and additions, and we try to find space for the most useful. Thanks to Mr Alcock.

FILER COMMAND LISTS

If you have added extra commands to your BEEBUG Filer Utilities, then the command list (when displayed from within the program) can scroll partially off the screen. One solution is to add a VDU14 to the procedure list. Alternatively, use the following line change to display commands in a two column format:

```
6340FORI=1TON:PRINTTAB(-40*
(I/2=INT(I/2)), (I-1)/2+2);c
om$(I):NEXTI
```

The line number is the same in the original versions of the Filer database, accounts and graphics programs.

Matthew Smith

Something very similar to this was incorporated in the final version of Filer as sold on the Filer disc. Other users of this database system may wish to amend their programs accordingly.

ONE FOR CHRISTMAS

I thought you might like to add the following alphametic to your list. Author, I'm afraid, unknown.

A + MERRY + XMAS = TURKEY

R.A.Jackson

You could try this out on the winning alphametics program (on last month's magazine disc/tape), but why not have some fun this Christmas and solve it by hand. Solution next time.

BEEBUG MEMBERSHIP

Send applications for membership renewals, membership queries and orders for back issues to the address below. All membership fees, including overseas, should be in pounds sterling drawn (for cheques) on a UK bank.

MEMBERSHIP SUBSCRIPTION RATES

£ 7.50 - 6 months (5 issues) UK only
 £14.50 - 1 year (10 issues) UK, BFPO, Ch.I
 £20.00 - Rest of Europe & Eire
 £24.50 - Middle East
 £27.00 - Americas & Africa
 £29.00 - Elsewhere

BACK ISSUE PRICES

Volume	Magazine	Cassette	5"Disc	3.5"Disc
1	£0.40	£1.00	-	-
2	£0.50	£1.00	£3.50	-
3	£0.70	£1.50	£4.00	-
4	£0.90	£2.00	£4.50	£4.50
5	£1.20	£2.50	£4.75	£4.75
6	£1.30	£3.00	-	-

All overseas items are sent airmail. We will accept official UK orders for subscriptions and back issues, but please note that there will be a £1 handling charge for orders under £10 which require an invoice. Note that there is no VAT in magazines.

FURTHER DISCOUNTS

We will allow you a further discount:

Five or more: deduct £0.50 from total
 Ten or more: deduct £1.50 from total
 Twenty or more: deduct £3.50 from total
 Thirty or more: deduct £5.00 from total
 Forty or more: deduct £7.00 from total

POST AND PACKING

Please add the cost of p&p:

Destination	First Item	Second Item
UK, BFPO + Ch.I	40p	20p
Europe + Eire	75p	45p
Elsewhere	£2	85p

BEEBUG
 Dolphin Place, Holywell Hill, St.Albans,
 Herts. AL1 1EX
 Tel. St.Albans (0727) 40303

Manned Mon-Fri 9am-5pm
 (24hr Answerphone for Connect/Access/Visa orders and subscriptions)

BEEBUG MAGAZINE is produced by BEEBUG Ltd.

Editor: Mike Williams
 Assistant Editor: Kristina Lucas
 Production Assistant: Yolanda Turuelo
 Membership secretary: Mandy Mileham
 Editorial Consultant: Lee Calcraft
 Managing Editor: Sheridan Williams

All rights reserved. No part of this publication may be reproduced without prior written permission of the Publisher. The Publisher cannot accept any responsibility whatsoever for errors in articles, programs, or advertisements published. The opinions expressed on the pages of this journal are those of the authors and do not necessarily represent those of the Publisher, BEEBUG Limited

CONTRIBUTING TO BEEBUG PROGRAMS AND ARTICLES

We are always seeking good quality articles and programs for publication in BEEBUG. All contributions used are paid for at up to £40 per page, but please give us warning of anything substantial that you intend to write. A leaflet 'Notes to Contributors' is available on receipt of an A5 (or larger) SAE.

Please submit your contributions on disc or cassette in machine readable form, using "View", "Wordwise" or other means, but please ensure an adequate written description is also included. If you use cassette, please include a backup copy at 300 baud. In all communication, please quote your membership number.

BEEBUG Ltd (c) 1987

Magazine Disc/Cassette

DECEMBER 1987 DISC/CASSETTE CONTENTS

SNOWFLAKES - a highly seasonal and graphical program for the December issue.

EXPLORING ASSEMBLER (PART 6) - a Basic program which simulates and demonstrates the use of assembler logic instructions.

PROCEDURE AND FUNCTION ANALYSER - provides a detailed cross-reference listing of all the functions and procedures in a Basic program.

PERSONAL APPOINTMENTS DIARY - month to view, or day to view, just the job for the new year.

BEEBUG WORKSHOP

THANKS FOR THE MEMORY - program demonstrating the use of indirection operators to save memory.

MODULAR ASSEMBLER - a utility for machine code enthusiasts for assembling long programs.

LOAD AND GO WITH BASIC PROGRAMS - a short utility allowing Basic programs to be *RUN just like machine code.

BARRY CHRISTIE VISUALS

TRENCH ANIMATION - another striking visual animation from our expert.

THE MASTER SERIES

ADFS SINGLE DRIVE BACKUP - a fast and efficient means to back up single-drive ADFS systems on the Master and Compact.

TRIVIA QUIZ - our Christmas game, a version of the popular trivia style quiz to provide fun for all the family.

MAGSCAN - Bibliography data for this issue of BEEBUG

Pitfall Pete - ten completely new screens for this highly popular game published in Vol.5 No.7.

All this for **£3 (cassette)**, **£4.75 (5" & 3.5" disc)** + 50p p&p. PLEASE STATE DISC TYPE
Back Issues (5.25" disc since Vol.3 No.1, 3.5" disc since Vol.5 No.1, tapes since Vol.1 No.10) available at the same prices.

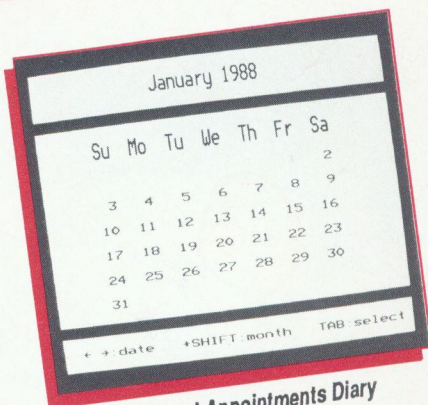
SUBSCRIPTION RATES
6 months (5 issues)
12 months (10 issues)

	5" Disc	3.5" Disc	Cassette
UK ONLY			
	£25.50	£25.50	£17.00
	£50.00	£50.00	£33.00

	5" Disc	3.5" Disc	Cassette
OVERSEAS			
	£30.00	£30.00	£20.00
	£56.00	£56.00	£39.00

Prices are inclusive of VAT and postage as applicable. Sterling only please.

Cassette subscriptions can be commuted to a 5.25" or 3.5" disc subscription on receipt of £1.70 per issue of the subscription left to run. All subscriptions and individual orders to:
BEEBUG, Dolphin Place, Holywell Hill, St.Albans, Herts. AL1 1EX.



Personal Appointments Diary

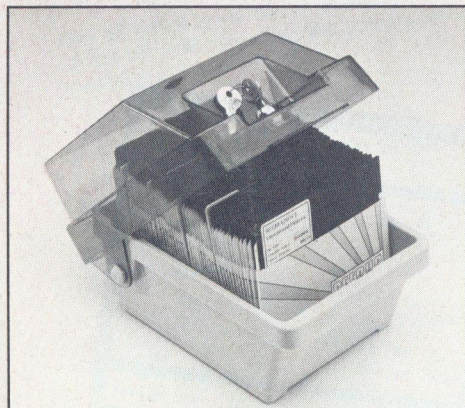
Filename: PF (6/52 bytes)			
No.	LINE NAME	BYTES	%
1	10 PROGRAM	496	7.3
2	1000 PROCinit	253	3.7
3	1090 PROCmenu1	478	7.1
4	1230 PROCtitle	108	1.6
5	1270 PROCfindDEFs	452	6.7
6	1410 PROCkeybytes	317	4.7
7	1510 PROCintable	207	3.1
8	1570 FNitemname	165	2.4
9	1620 PROCbuildmatrix	311	4.6
10	1730 PROCinMatrix	102	1.5
11	1760 FNnum	123	1.8
12	1810 PROCmenu2	240	3.6
13	1880 PROCmenu3	386	5.7
14	1990 PROCprinter	286	4.2
15	2060 PROCscreenDEFs	127	1.9
16	2110 PROCscreenhead	226	3.3
17	2170 PROCscreenline	162	2.4
18	2220 PROCgetinfo	128	1.9
19	2260 PROCscreencells	368	5.4
20	2360 FNname	82	1.2
21	2390 PROCprintDEFs	199	2.9

Procedure/Function Analyser

BEEBUG Discs - The Ultimate in Quality & Reliability



SPECIAL CHRISTMAS OFFER



50 discs with free lockable storage box

Prices shown are members prices and include VAT.

These offers represent £10 off our normal prices and are only available until 31/1/88. BEEBUG discs are manufactured to the highest specifications and are fully guaranteed.

50 Single Sided 40T Double Density Discs **Only £35.60**

50 Double Sided 80T Quad Density Discs **Only £39.90**

Our Guarantee

We confidently offer a lifetime data guarantee and will replace any disc with which you encounter problems. We have found that the standard of quality control at the factory makes this necessity very rare.

40 Track Single Sided Double Density

	Price	Members Price	Order Code
10	£9.90	£9.40	0657
25	£23.00	£21.85	0661
50	£37.50	£35.60	0665

80 Track Double Sided Quad Density

	Price	Members Price	Order Code
10	£10.90	£10.35	0660
25	£29.00	£27.55	0664
50	£42.00	£39.90	0668

Please send me _____ (qty) _____ (stock code) at £_____ (unit price)

UK post 10 £1, 25/50 £3.75. Overseas send same price inc. UK post & VAT

I enclose a cheque for £_____ / Please debit my Access/Visa card £_____

Expiry date:

Name _____ Memb No. _____

Address _____

BEEBUG

Dolphin Place,
Holywell Hill,
St. Albans,
Herts.
AL1 1EX

☎ (0727) 40303